

Test de recrutement : programmation

Paul-Antoine BISGAMBIGLIA

Faculty of Science

UMR-CNRS-6134

University of Corsica – 20250 Corte

bisgambiglia@univ-corse.fr

Mots-clés : Algorithme, programmation

Est-ce que je vous recrute ! 2 points par exercice fonctionnel qui répond à la consigne. 4 points si c'est terminé en 2h, 3 points en 2h30, 2 points en 3h.

Déposez vos codes sur l'ent.

Exercice 1 :

Etape 1 : le but du code

Nous souhaitons avoir une interface sous forme de grille NxN (6x6 par exemple).

Dans chaque case, nous devons pouvoir afficher un objet (une image voire un caractère comme A par exemple).

	A		

Etape 2 : génération aléatoire de la grille

Nous souhaitons afficher la grille avec K objets ($K < N$).

$N = 4$, $K = \text{random}(1,4)$ par exemple 3

		A	
	A		
A			

Etape 3 : algo à chercher (algo de peuplement)

Nous souhaitons pouvoir tester un algo qui doit trouver tous les K objets et placer à côté (à droite ou à gauche) un nouvel objet de tel sorte que chaque objet est un et unique voisin.

		A	A
	A	A	

A	A		

Mauvais cas, 2 A on 2 voisins :

A	A	A	A

Etape 4 : complexification

Modifier la fonction de l'étape deux pour ajouter des casses bloquées ($P < N/2$).

Vérifier si votre algo de peuplement fonctionne, si non modifiez le.

		A	
	A	P	
	P		
A			

Une solution :

		A	A
A	A	P	
	P		
A	A		

Attention il peut y avoir des cas sans solution, ça sera à indiquer en retour de votre fonction.

Exercice 2 :

Dans une chaîne, détecter la plus longue chaîne de caractères composés de caractères distincts.

Par exemple : "abcdemo" est la plus longue chaîne de caractères distincts de "abcdemoderneancien"

Exercice 3 :

Concevoir un algorithme pour trouver tous les caractères communs à deux listes triées.

Par exemple, pour les listes a, e, e, e et b, b, c, e, e, g, la sortie doit être de e, e.

Exercice 4 :

Diviser un tableau de nombre en deux de manière à ce que la différence entre les deux tableaux soit la plus petite possible.

Exercice 5 :

Ecrire une fonction qui calcule la longueur moyenne des mots d'un texte.

```
sentence1 = "Même les phrases avec des caractères de la langue  
française peuvent être utilisées."
```

```
print(average_words_length(sentence1)) [Output] => 5.38
```

Exercice 6 :

Nous souhaitons inverser un entier (positif ou négatif), c'est-à-dire notre fonction prend en entrée un entier -6523 par exemple et retourne en sortie l'entier inversé -3256.

```
# Afficher les résultats de notre fonction pour 2020 et -9430
```

```
print(reverse_int(2020)) => 202
```

```
print(reverse_int(-9430)) => -349
```

Exercice 7 :

Retrouver dans une liste d'entiers, tous les triplets pythagoriciens possibles qui y sont. Pour rappel, un triplet pythagorien respecte le théorème suivant : $a^2 + b^2 = c^2$.

Prenons l'exemple suivant : nous avons la liste [0, 3, 6, 1, 2, 4, 5]. Notre fonction renvoie la liste des triplets possibles : [(3, 4, 5)] car $9 + 16 = 25$.

Exercice 8 :

Nous avons un mot et nous voulons savoir quel est le premier caractère unique de ce mot, c'est-à-dire la lettre qui ne se répète pas dans le mot et la première.

```
print(first_unique_character('coronavirus')) [Output] ('c', 0)
```

```
print(first_unique_character('Europe')) [Output] ('u', 1)
```