

Sujets tirés d'Agro-Véto

Mathématiques et informatique

Les exercices proposés sont extraits d'épreuves du concours Agro-Véto. Ils sont recopiés ici sans aucune modification.

1 Liste de booléens

Dans cet exercice, on considère des listes de booléens, c'est-à-dire des listes ayant pour éléments uniquement des valeurs `True` et `False`. Par exemple les listes `[True, False, True]` et `[False, False]` sont des listes de booléens.

1. Écrire une fonction `nb_vrai` prenant en entrée une liste de booléens et renvoyant le nombre d'occurrences de la valeur `True`. Par exemple `nb_vrai([False, True, True])` doit renvoyer 2, car la valeur `True` apparaît deux fois dans la liste `[False, True, True]`.
2. Écrire une fonction `bloc_vrai` prenant en entrée une liste de booléens, et renvoyant :
 - `True` si la valeur `True` est présente dans la liste et que tous les `True` sont côte-à-côte, dans le même bloc.
 - `False` sinon.

Ainsi, `bloc_vrai([False, True, True, True, False, False])` doit renvoyer `True` car il y a un unique bloc de 3 `True`.

Autre exemple : `bloc_vrai([False, True, True, True, False, False, True, True, False])` doit renvoyer `False` car il y a deux blocs de `True` : un premier de longueur 3, un second de longueur 2.

2 Points fixes dans une liste

On considère un plateau de n cases, numérotées de 0 à $n - 1$. Sur chaque case, se trouve un numéro de case (le même numéro peut être inscrit sur plusieurs cases différentes). Par exemple, sur le plateau suivant, on a écrit 2 sur la case d'indice 0, et on a écrit 4 sur la case d'indice 1 et on a écrit 3 sur la case d'indice 3.

2	4	1	3	0
---	---	---	---	---

Informatiquement, on représente un plateau de taille n par une liste de taille n . La k -ième case de la liste contient le numéro inscrit sur la k -ième case. Ainsi, le plateau de l'exemple est représenté par la liste `[2, 4, 1, 3, 0]`.

1. On appelle point fixe une case sur laquelle est inscrit son propre numéro. Ainsi, sur l'exemple il y a un unique point fixe, la case 3 (car il est écrit 3 sur la case d'indice 3). Écrire une fonction `compter` qui prend en entrée une liste représentant un plateau et qui renvoie le nombre de points fixes (la fonction renvoie 0 s'il n'y a aucun point fixe).

On place un pion sur la case zéro. Ensuite, on fait avancer le pion comme suit : on regarde le numéro inscrit sur la case sur laquelle est le pion, puis on déplace le pion sur la case portant ce numéro. Sur l'exemple, on va déplacer le pion sur la case d'indice 2, puis la case d'indice 1, puis la case d'indice 4, puis la case d'indice 0, puis la case d'indice 2, etc...

2. Écrire une fonction `avancer` qui prend en entrée la liste représentant le plateau de jeu et un entier naturel p et qui donne le numéro de la case sur laquelle arrive le pion au bout de p étapes.

Par exemple, `avancer([2, 4, 1, 3, 0], 1)` renvoie 2 et `avancer([2, 4, 1, 3, 0], 3)` renvoie 4.

3 Monotonie

1. Écrire une fonction `strict_croissante` prenant une liste d'entiers en paramètre et renvoyant `True` si elle est strictement croissante et `False` sinon.

Par exemple, `strict_croissante([1, 7, 99])` renvoie `True`.

Mais `strict_croissante([1, 7, 5])` et `strict_croissante([1, 7, 7])` renvoient `False`.

2. Écrire une fonction `strict_monotone` prenant une liste d'entiers en paramètre et renvoyant `True` si elle est strictement monotone et `False` sinon.

Par exemple, `strict_monotone([5, 2, 1])` renvoie `True`.

Mais `strict_monotone([1, 2, 1])` renvoie `False`.