

Exercice : Listes

Exercice 1

La suite de Fibonacci est une suite dont chaque valeur s'obtient en ajoutant les deux valeurs qui la précèdent, avec les nombres 0 et 1 comme valeurs initiales.

Soit **t** la liste des premiers nombres de la suite de Fibonacci :

$$t = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]$$

- 1) Saisir la liste **t** donnée ci-dessus.
- 2) Afficher sur 3 lignes, avec une seule instruction **print**, la longueur de ce tableau, le premier élément et son dernier élément. Vous devez obtenir :

nombre de valeurs : 12
premier élément : 0
dernier élément : 89

- 3)
 - a) Créer un second tableau **t1** de la même dimension que **t** dont toutes valeurs sont 0.
 - b) En utilisant une boucle **for**, compléter le tableau **t1** avec les valeurs du tableau **t** rangées dans l'ordre inverse. On doit obtenir :

[89,55,34,21,13,8,5,3,2,1,1,0]

- 4) Créer un tableau dont les valeurs sont la somme des valeurs des tableaux **t** et **t1** de même indice.
- 5) Créer la fonction **terme_suivant** qui prend en paramètre la liste des nombres de la suite de Fibonacci et ajoute le terme suivant en fin de liste.

Exercice 2

- 1) Créer une fonction **tabalea** qui renvoie un tableau avec des entiers choisis aléatoirement compris entre 1 et 100 et dont la dimension sera passée en paramètre.
*Par exemple : **tabalea**(50) renvoie un tableau avec 50 entiers aléatoirement pris entre 1 et 100.*
- 2) Créer une fonction **tabmax** qui prend en paramètre un tableau et qui renvoie son plus grand élément et son indice.
- 3) Créer une fonction **tabin** qui prend en paramètres un nombre entier et un tableau et renvoie **True** si le nombre est présent dans le tableau et **False** dans le cas contraire.
- 4) Créer une fonction **tabmode** qui prend en paramètre un tableau et renvoie la valeur dominante, c'est à dire la valeur du tableau la plus représentée, et le nombre de fois que celle-ci est présente.
- 5) Créer la fonction **permute** qui prend en paramètre la liste de nombres et la modifie en décalant toutes les valeurs de 1 rang vers la droite. La première position, laissée libre, sera occupée par la dernière valeur du tableau qui a été éjectée.

Exercice 3

- 1) Créer un tableau de 8 valeurs, chaque valeur étant un fruit commençant par une lettre différente! Trier le tableau avec la fonction **sort**.
- 2) Créer une fonction **échange** prenant en paramètre un tableau et deux indices **i** et **j** et qui renvoie le tableau avec les valeurs d'indices **i** et **j** permutées.
- 3) Créer une fonction **mélange** qui prend en paramètre un tableau et mélange aléatoirement les valeurs de ce tableau. On pourra utiliser la fonction **échange** à plusieurs reprises avec des indices choisis aléatoirement.

Exercice 4

Créer les tableaux suivants avec la méthode par compréhension :

- 1) Liste des nombres entiers de 0 à 12 inclus
- 2) Liste des nombres entiers pairs de 0 à 16 inclus
- 3) Liste des nombres impairs de 1 à 15 inclus
- 4) Liste des carrés des nombres entiers de 1 à 5 inclus
- 5) Liste des lettres de l'alphabet en majuscule
- 6) `[[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]`
- 7) `[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]]`

Exercice 5

- 1) Créer un tableau `t` de dimension aléatoire `n` dont les valeurs sont des nombres entiers consécutifs inférieurs à 20. Par exemple, si `n=4`, alors `t=[0,1,2,3,4]`
- 2) Créer une fonction **prolonge** qui ajoute au tableau `t` de nombres entiers rangés par ordre croissant, le nombres entiers consécutifs jusqu'à 20.
- 3) Créer une fonction **miroir** qui ajoute au tableau `t` de nombres entiers rangés par ordre croissant, les nombres entiers décroissants jusqu'à 0 sans aucune répétition.

Exercice 6

- 1) Créer deux tableaux **jours** et **mois** regroupant toutes les valeurs possibles.
- 2) Créer la fonction **affiche_janvier** affichant tout le mois de janvier commençant par un lundi 1 janvier.
- 3) Modifier la fonction **affiche_janvier** prenant en paramètre le premier jour du mois (lundi, mardi, mercredi,...) et affichant le mois en conséquence.
- 4) Créer la fonction **affiche_mois** prenant comme paramètre un mois et le premier jour du mois et affichant le mois. On prendra en compte le nombre de jours dans le mois.
- 5) Écrire une fonction **est_bissextile** prenant en paramètre l'année et retourne **True** si elle est bissextile ou **False** dans le cas contraire. On rappelle qu'une année est bissextile si elle est un multiple de 4 mais pas un multiple de 100, ou si elle est un multiple de 400.
- 6) Modifier la fonction **affiche_mois** en ajoutant comme paramètre l'année et affichant le mois de février avec 28 ou 29 jours.
- 7) Modifier la fonction **affiche_mois** pour qu'elle retourne un tableau des dates demandées.

Exercice 7

- 1) Créer une fonction **nouvelleGrille** qui crée un tableau de tableaux de dimension 10×10 , c'est à dire de 10 lignes et 10 colonnes initialisée avec des zéros.
Créer la variable **maGrille** qui est une grille 10×10 de zéros.
- 2) Créer une fonction **alimente** qui remplit la grille avec des nombres entiers aléatoirement choisis entre 0 et 200. Alimenter la variable `ma_grille`.
- 3) Créer une fonction **tabmax** qui renvoie la plus grande valeur d'un tableau (voir exercice 2).
- 4) Créer une fonction **transpose** qui prend en paramètre une grille et renvoie une grille dont les lignes sont les colonnes de la grille saisie en paramètre.
- 5) Créer une fonction **maxligne** qui renvoie un tableau contenant la valeur maximale de chaque ligne de la grille. Penser à utiliser la fonction `tabmax`.
- 6) Créer une fonction **maxicolonne** qui renvoie un tableau contenant la valeur maximale de chaque colonne de la grille. Penser à utiliser la fonction `tabmax` et la fonction `transpose`.

Exercice 8

Le Verger est un jeu de société coopératif créé en 1986. Ce jeu contient :

- un plan de jeu présentant 4 arbres fruitiers (prunier, pommier, poirier, cerisier) ainsi qu'un corbeau au centre.
- 40 fruits (10 pour chaque arbre du plan de jeu) en bois à disposer sur chacun des arbres.
- 4 paniers pour les joueurs (1 panier par joueur)
- Un puzzle corbeau de 9 pièces que l'on reconstitue lors de la partie.
- Un dé dont les faces représentent les 4 arbres, le panier et le corbeau.



Règles du jeu

Le but du jeu est de récupérer tous les fruits avant d'avoir reconstitué le puzzle du corbeau.

Un joueur lance le dé :

- Si le dé tombe sur un arbre, le joueur prend le fruit correspondant et le met dans son panier. S'il n'y a plus de fruit sur l'arbre, le joueur passe son tour.
- Si le dé tombe sur le panier, le joueur prend 2 fruits de son choix.
- Si le dé tombe sur le corbeau, il place une pièce du puzzle sur le corbeau.

Le gagnant

Les joueurs gagnent tous ensemble s'ils ont réussi à cueillir tous les fruits de tous les arbres. Les joueurs perdent ensemble si le puzzle du corbeau est reconstitué avant d'avoir cueilli tous les fruits.

Programmation

Écrire un programme qui permet d'effectuer une partie et affiche le gagnant.

Exercice 9

Le jeu du loto se joue avec des grilles différentes composées de 15 numéros compris entre 1 et 90 (inclus). Au cours d'une partie, des numéros sont tirés au hasard. Le premier à avoir une grille complète est le gagnant du lot principal.

On donne un exemple de grille ci-dessous :

2		23		43	51			81
	16	24		48		61	77	
4	19		38			64		88

Écrire un programme produisant une carte de loto en respectant les contraintes suivantes :

- Une carte contient 9 colonnes et 3 lignes.
- Il y a sur la carte 15 numéros différents choisis parmi les nombres de 1 à 90.
- Chaque ligne contient 5 numéros (et donc 4 espaces vides).
- Il y a toujours au moins un numéro par colonne.
- Il peut y avoir 3 numéros dans une colonne, mais seulement dans la colonne 8.
- La colonne 0 contient les numéros de 1 à 9.
- La colonne 1 contient les numéros de 10 à 19.
- La colonne 2 contient les numéros de 20 à 29.
- ...
- La colonne 8 contient les numéros de 80 à 90.