Devoir surveillé n°1

Durée : 1 heure et 45 minutes, calculatrices, ordinateurs et documents interdits

I. Exercice sur les bases de données

Dans cet exercice, nous étudions une base de données enregistrant les données d'un service de *streaming* musical. La base de données contient de nombreuses informations sur les artistes, albums et morceaux de musique disponibles. Lorsqu'un utilisateur écoute un morceau de musique, les informations de son écoute sont aussi enregistrées dans la base de données. On considérera les tables suivantes.

- ▷ La table `artistes`, qui contient les attributs suivants :
  - `id_artiste` : identifiant de l'artiste (ou du groupe) ;
  - `nom` : nom de scène de l'artiste (ou du groupe).
- ▷ La table `albums`, qui contient les attributs suivants :
  - `id_album` : identifiant de l'album ;
  - `titre` : titre de l'album ;
  - `annee_sortie` : année de sortie de l'album ;
  - `id_artiste` : identifiant de l'artiste auteur de l'album.
- ▷ La table `morceaux`, qui contient les attributs suivants :
  - `id_morceau` : identifiant du morceau musical ;
  - `titre` : titre du morceau ;
  - `duree` : durée du morceau, exprimée en secondes ;
  - `id_album` : identifiant de l'album auquel appartient le morceau.
- ▷ La table `ecoutes`, qui contient les attributs suivants :
  - `id_ecoute` : identifiant de l'écoute ;
  - `id_morceau` : identifiant du morceau écouté ;
  - `duree_ecoute` : durée d'écoute du morceau, exprimée en secondes ;
  - `date_ecoute` : date à laquelle a été écoutée le morceau.

Une annexe se trouve à la fin de l'exercice. Il s'agit d'exemples dont l'objectif est d'aider à la compréhension de la base de données. Leur contenu ne doit pas être utilisé pour répondre aux questions !

1) Donner pour chaque table une clef primaire, et les éventuelles clefs étrangères s'y trouvant (ainsi que les clefs primaires auxquelles elles font référence).

Dans la suite de l'exercice, on répondra aux questions par des requêtes SQL.

2) Afficher toute la table `morceaux`.

3) Afficher les informations des albums sortis en 2025.

4) Afficher deux colonnes : la première colonne comportera les titres des albums, la seconde les titres des morceaux de musique de l'album correspondant.

5) Afficher les titres des morceaux écoutés en 2026.

6) Un artiste réenregistre parfois des morceaux. Dans la base de données, cela revient à avoir deux morceaux portant le même titre et provenant du même artiste.

Afficher la liste des noms des morceaux qui ont été réenregistrés.

7) Calculer le nombre total de morceaux de la base de données.

8) Pour chaque artiste, afficher le temps total d'écoute (exprimé en minutes) et le nombre d'écoutes.

9) Afficher les titres des albums dont la durée moyenne des morceaux est au moins de 5 minutes.

10) Un morceau est diffusable à la radio s'il dure au plus 4 minutes. Afficher les albums ayant au moins 5 morceaux diffusables à la radio.

11) Afficher titre du morceau de musique le plus écouté (en nombre total d'écoutes).

12) Afficher les titres des morceaux qui n'ont jamais été écoutés en entier.

Annexe

Les tableaux ci-dessous sont donnés à titre d'**exemples** ; ils contiennent des **extraits** des quatre tables. Leur contenu ne doit donc pas être utilisé pour répondre aux questions.

***Extrait** de la table* `artistes`

id_artiste	nom
...	...
1	"Daft Punk"
2	"Epica"
...	...

***Extrait** de la table* `albums`

id_album	titre	annee_sortie	id_artiste
...	...	...	...
10	"Discovery"	2001	1
11	"Aspiral"	2025	2
...	...	...	...

***Extrait** de la table* `morceaux`

id_morceau	titre	duree_morceau	id_album
...	...	...	...
101	"One More Time"	320	10
102	"Arcana"	302	11
...	...	...	...

***Extrait** de la table* `ecoutes`

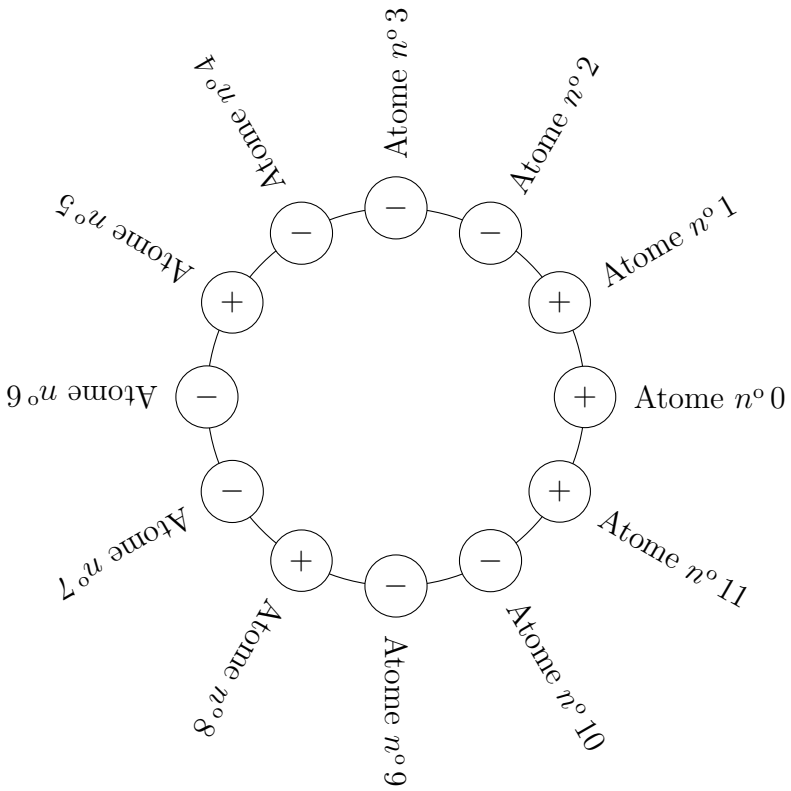
id_ecoute	id_morceau	date_ecoute	duree_ecoute
...	...	...	...
1001	101	"2026-10-16"	317
1002	113	"2022-02-29"	111
...	...	...	...

II. Algorithmique : le modèle d'Ising

On considère un réseau de n atomes métalliques disposés en cercle, plongés dans un champ magnétique (voir figure ci-dessous). Chaque atome a un moment magnétique, qui peut posséder deux orientations magnétiques (ou spin) : une orientation positive (+), une orientation négative (−). Ces orientations seront représentées par une liste $S = [s_0, \ldots, s_{n-1}]$: $s_i = 1$ si l'atome $n^o\, i$ a une orientation +, et $s_i = -1$ si l'atome $n^o\, i$ a une orientation −.
Par exemple, la liste

$$S = [1, 1, -1, -1, -1, 1, -1, -1, 1, -1, -1, 1]$$

représente le réseau suivant (ici, $n = 12$). On remarquera que cette liste n'est qu'un exemple, et ne doit pas être utilisée dans les réponses.



On considère, dans les notations mathématiques qui suivent, la liste S périodisée : $s_n = s_0$, $s_{-1} = s_{n-1}$ et, de manière générale, pour tout $i \in \mathbb{Z}$: $s_i = s_{n+i}$.
Notamment, si $i \in \mathbb{Z}$ et si r est le reste de la division euclidienne de i par n : $s_i = s_r$. On rappelle que ce reste de division euclidienne peut s'obtenir en Python par l'opérateur `%`.

On supposera que le système est à une température T, valeur que l'on suppose être déjà affectée à une constante `T`. On rappelle que la constante de Boltzmann est $k_B \approx 1,38 \times 10^{-23} J.K^{-1}$.

L'aimentation totale du réseau est

$$M(S) = \alpha(T) \sum_{i=0}^{n-1} s_i,$$

où $\alpha(T)$ est une constante dépendant de la température, exprimée en $A.m^{-1}$, déterminée expérimentalement. Ici, on suppose qu'une fonction `alpha` est déjà fournie, qui prend en argument le paramètre `T`, et qui renvoie la valeur de la constante α(`T`).

Dans vos réponses, vous ne devrez pas utiliser de fonction effectuant un calcul avancé sur les listes (exemple : `max`, `sum`, `in`, `==` entre deux listes). Vous pourrez utiliser la fonction `len` et la méthode `.copy()` .

13) Écrire une fonction `aimentationTotale(S)`, qui renvoie la valeur de l'aimantation totale M du réseau décrit par la liste S.

On considère qu'un atome n'intéragit qu'avec ses deux voisins. On considère qu'il existe une constante de couplage J (homogène à une énergie) et que le réseau est soumis à un champ magnétique extérieur H. L'énergie de la configuration est alors

$$E(S) = H \cdot M(S) - J \sum_{i=0}^{n-1} s_{i-1} s_i.$$

Dans toute la suite du sujet, on suppose que les valeurs de J et de H ont déjà été affectées à deux variables, respectivement `J` et `H`.

14) Écrire une fonction `energie(S)`, qui renvoie la valeur d'énergie $E(S)$ du réseau décrit par la liste S.

15) Écrire une fonction `rupture(S)`, qui renvoie un entier i vérifiant $s_{i-1} \neq s_i$.
S'il n'en existe aucun, alors `rupture(S)` renverra un indice quelconque de $[\![0, n[\![$.

16) Écrire une fonction `plusGrandePlage(S)` renvoyant le plus grand nombre d'atomes consécutifs de même spin. Cette fonction renverra la valeur n si tous les atomes sont de même spin.

On va maintenant simuler l'évolution de ce système, à partir des règles suivantes.

- ▷ Les spins sont au départ tirés uniformément dans $\{-1, 1\}$, indépendamment les uns des autres.
- ▷ On réalise N fois l'opération suivante.
  - À chaque étape, on choisit un atome au hasard parmi les n atomes présents.
  - Si l'on note E l'énergie du système au début de cette étape, et E' l'énergie du système que l'on obtient en inversant le spin de l'atome choisi, on note $\Delta E = E' - E$.
  - On tire un nombre flottant r au hasard dans $[0, 1[$. Si

$$r \leqslant \exp\left(-\frac{\Delta E}{k_B T}\right),$$

    alors on change le spin de l'atome choisi, en son opposé. Sinon, on ne change pas le spin de l'atome choisi.

On rappelle que, dans la bibliothèque `random`, on dispose de deux fonctions de tirage aléatoire :

random : `random()` renvoie un nombre flottant tiré uniformément dans $[0, 1[$;

randrange : si `a` est un entier, `randrange(a)` renvoie un nombre entier tiré uniformément dans $[\![0, \mathtt{a}[\![$.

17) On suppose que, au début de l'étape, le réseau est décrit par la liste $S = [s_0, \ldots, s_{n-1}]$. On note i l'indice de l'atome choisi.
Montrer que $\Delta E = 2J(s_{i-1}s_i + s_i s_{i+1}) - 2H\alpha(T)s_i$.

18) Écrire une fonction `spinInitial()`, sans argument, qui renvoie 1 avec probabilité $\frac{1}{2}$, et -1 avec probabilité $\frac{1}{2}$.

19) Écrire une fonction `evolution(n, N)` renvoyant la liste décrivant le système obtenu après N étapes.

— FIN —