

Interrogation d'informatique n° 1 MP

vendredi 9 octobre 2026 (1h)



Exercice 1 (14 points)

On dispose de la table **Incidents** qui répertorie les incidents techniques survenus dans une entreprise. Les attributs sont les suivants :

1. **id** (clé primaire, entier) : Identifiant.
2. **titre** (texte) : Titre court décrivant l'incident (ex : *Panne serveur web*).
3. **categorie** (texte) : Type d'incident (ex : *Matériel, Logiciel, Réseau, Sécurité*).
4. **gravite** (entier) : Échelle numérique (ex : 1 à 5).
5. **statut** (texte) : *Signalé, En cours, Résolu*.
6. **priorite** (texte) : *Basse, Normale, Haute, Urgent*.
7. **cout_estime** (décimal) : Coût estimé de réparation (en euros).

Voici un extrait de la table :

id	titre	categorie	gravite	statut	priorite	cout_estime
1	Panne serveur web	Matériel	5	Signalé	Urgent	1250.00
2	Bug application RH	Logiciel	3	En cours	Haute	450.50
3	Connexion réseau instable	Réseau	2	Résolu	Normale	180.75
4	Fuite de données	Sécurité	4	Signalé	Urgent	3200.00
5	Imprimante hors service	Matériel	1	Résolu	Basse	95.20

On demande d'écrire une requête SQL pour chacune des questions suivantes.

1. Quels sont tous les incidents de *gravité supérieure ou égale à 4* qui sont encore *signalés* ?
2. Quels sont les *titres* et *gravités* des incidents de catégorie *Sécurité* ou *Réseau* ?
3. Quels sont les *titres* et *coûts estimés* des incidents de priorité *Urgent*, classés par coût décroissant ?
4. Combien d'incidents y a-t-il par *catégorie* ?
5. Quel est le *coût estimé moyen* des incidents *résolus* ?
6. Quelles sont les *catégories* d'incidents pour lesquelles le *coût estimé total par catégorie* dépasse *1000 euros* ?
7. Quels sont les *2 incidents* les plus coûteux (par *cout_estime*) parmi ceux de catégorie *Matériel*, en excluant le premier résultat ?

Exercice 2 (2 points)

Écrire une fonction `transforme(T)` qui prend en entrée une liste de n -listes d'entiers de taille n (le format est donc carré) et qui renvoie la liste de liste transformée comme suit : à la place d'un entier pair, on met un 0, et à la place d'un entier impair, on met un 1.

Exercice 3 (4 points)

1. Proposer une fonction Python `maxi` prenant en argument une liste d'entiers naturels L et renvoyant le maximum des entiers de cette liste.

On n'utilisera pas de fonction spécifique de python déterminant ce maximum.

2. Écrire une fonction `ind` prenant en argument une liste d'entiers naturels L et renvoyant la liste des indices $[i_1, \dots, i_r]$ avec $i_1 < \dots < i_r$ telle que pour tout $k \in \llbracket 1, r \rrbracket$, $L[i_k]$ soit non nul.

Par exemple si $L = [0, 1, 3, 0, 7]$, alors `ind(L)` renvoie $[1, 2, 4]$.

3. Écrire une fonction `nb_oc` prenant comme argument une liste d'entiers naturels L et renvoyant la liste T de longueur $M = \text{maxi}(L)+1$ où, pour tout $i \in \llbracket 0, M \rrbracket$, $T[i]$ est le nombre d'occurrences de l'entier i dans la liste L .

Par exemple, si $L = [3, 1, 4, 1, 5]$, alors $T = [0, 2, 0, 1, 1, 1]$.

On pourra utiliser la fonction `maxi`.

4. Soit L une liste d'entiers naturels.

- (a) Déterminer le nombre de fois, noté n , où la liste L est parcourue lors de l'exécution de `nb_oc(L)`.
- (b) On veut que ce nombre n soit indépendant de $M = \text{maxi}(L)+1$.
Si ce n'est pas le cas, modifier la fonction `nb_oc` afin de respecter cette condition.

5. Soit A une liste d'entiers naturels. On définit la suite de Robinson $(L_n)_{n \in \mathbb{N}}$ associée à la suite A par récurrence comme suit :

- $L_0 = A$.
- Si L_n est construite, alors :
 - on détermine $T_n = \text{nb_oc}(L_n)$.
 - on détermine $I_n = \text{ind}(T_n)$.
 - si $I_n = [i_1, \dots, i_r]$, alors $L_{n+1} = [T[i_r], i_r, \dots, T[i_1], i_1]$.

Par exemple si $A = [4, 4, 1, 2]$, alors :

- $L_0 = [4, 4, 1, 2]$
- $L_1 = [2, 4, 1, 2, 1, 1]$ (il y a deux « 4 », un « 2 » et un « 1 » dans la liste L_0)
- $L_2 = [1, 4, 2, 2, 3, 1]$ (il y a un « 4 », deux « 2 » et trois « 1 » dans la liste L_1)

- (a) On donne $A = [2, 0, 4, 1, 3, 3, 2, 3, 1, 1]$. Déterminer L_3 et L_{2024} .
 - (b) On donne $B = [2, 4, 1, 1, 1, 2]$. Si l'on suppose que $L_1 = B$, donner toutes les solutions possibles pour L_0 .
 - (c) On donne $C = [2, 4, 1, 0]$. Si l'on suppose que $L_1 = C$, donner toutes les solutions possibles pour L_0 .
 - (d) (*bonus*) Proposer une fonction `rob(A, n)` qui prend en arguments une liste d'entiers naturels A et un entier naturel n et qui renvoie l'élément L_n de la suite de Robinson associée à A .
-