

Concours Blanc : épreuve d'Informatique

PTSI A et B Lycée Eiffel

28 mai 2014

Durée : 2H.

Des histoires de voitures.



Première partie : Bases de données.

Un concessionnaire automobile dispose de plusieurs garages situés dans une même région. Il dispose d'une base de données lui permettant de visualiser rapidement l'état de ses stocks dans ses différents garages. Cette base de données est constituée de trois tables :

- une table **garages** contenant les attributs suivants : *n°id* (nombre entier), *ville* (chaîne de caractères), *adresse* (chaîne de caractères), *nombre d'employés* (entier).
- une table **voitures** contenant les attributs suivants : *nom du modèle* (chaîne de caractères), *marque* (chaîne de caractères), *nombre de places* (entier), *vitesse maximale* (entier).
- une table **vente** contenant les attributs suivants : *n°id* (entier, représente un numéro de garage), *nom du modèle* (chaîne de caractères, représente un modèle de voiture), *prix* (entier), *occasion* (booléen, prend la valeur True s'il s'agit d'une voiture d'occasion, False sinon).

1. Préciser, pour chacune des tables, quel(s) attribut(s) sont susceptibles de constituer une clé primaire raisonnable de la table.
2. Écrire une requête dans le langage de l'algèbre relationnelle permettant d'obtenir la liste de tous les modèles de voitures ayant au moins 5 places. Écrire la requête SQL correspondante.
3. Écrire une requête SQL permettant d'obtenir le prix moyen des voitures vendues dans le garage ayant pour *n°id* le nombre 3.
4. Écrire une requête dans le langage de l'algèbre relationnelle permettant de faire la liste des marques de voitures disponibles dans les garages de la ville de Bordeaux. Traduire cette requête en langage SQL.
5. Écrire une requête SQL permettant d'ajouter à la table **vente** un attribut *annee de construction*, contenant une variable entière.
6. Écrire alors une requête SQL permettant d'obtenir la liste des voitures d'occasion construites après 2009, en affichant également l'adresse du garage correspondant.

Deuxième partie : Programmation en Python.

Notre concessionnaire a extrait pour l'analyser plus en détail un fichier texte contenant les noms des modèles de ses voitures, ainsi que leur prix, sous le format suivant :

```
Extrait du fichier tutures.txt
```

```
Peugot308;12500  
FerrariF40;500000
```

Chaque ligne du fichier contient donc un nom de modèle suivi d'un entier, les deux étant séparés par un point virgule. Pour cette partie du problème, on pourra utiliser la méthode Python *split*, qui s'applique à une chaîne de caractères et prend comme argument un caractère, et découpe la chaîne en plusieurs chaînes, en effectuant une séparation à chaque fois que le caractère spécifié en argument apparaît dans la chaîne. Ainsi,

```
chaine='abracadabra'  
chaine.split(a)
```

va découper la variable chaîne en quatre chaînes de caractères respectivement égales à 'br', 'c', 'd', et 'br'. Pour toutes les questions de cette partie, il est interdit d'utiliser d'autres fonctions ou méthodes Python que celles évoquées par l'énoncé, ainsi que les fonctions **range** et **len**.

1. Rappeler quels sont les différents modes d'ouverture d'un fichier texte à l'aide de Python, et les différences entre ces modes.
2. Rappeler le nom de la méthode permettant d'ajouter un élément à la fin d'une liste en Python.

3. Écrire une séquence d'instructions en Python permettant d'effectuer les opérations suivantes sur le fichier **tutures.txt** : ouvrir le fichier, créer deux listes **modeles** et **prix**, et stocker à l'intérieur de ces deux listes tous les modèles et les prix contenus dans le fichier texte (on pourra écrire une boucle commençant par « for ligne in f »).
4. On suppose pour la suite de cette partie que les listes **modeles** et **prix** ont été créées, et numérotées de façon cohérente (le prix du modèle situé dans la case numéro i de la liste modèle est dans la case numéro i de la liste prix).
Écrire une fonction Python prenant comme argument une liste, et retournant le plus petit élément situé dans cette liste.
5. Décrire pas à pas le fonctionnement de votre programme (en précisant la valeur de chaque variable après chaque étape de la boucle) si on l'applique à la liste [7, 5, 12, 1, 8].
6. Écrire une fonction Python prenant comme argument une liste et retournant la valeur moyenne de ses éléments.
7. En déduire une suite d'instructions en Python permettant d'afficher la liste des modèles disponibles dont le prix est inférieur à la moitié du prix moyen de tous les véhicules.
8. On rappelle que l'écart-type d'une série de valeurs est défini par la formule $\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - m)^2}$,
où n est le nombre total de valeurs, et m la moyenne de la série (les valeurs étant notées x_i).
Écrire une fonction calculant l'écart-type des éléments d'une liste (on réutilisera la fonction écrite plus haut pour la moyenne).
9. On souhaite écrire une fonction Python permettant d'afficher non plus le prix minimal parmi toutes les voitures, mais le nom de modèle correspondant. Décrire le fonctionnement d'un tel algorithme, puis écrire une suite d'instructions le réalisant.
10. Que va afficher le programme précédent si plusieurs voitures ont le même prix minimal ? Modifier le programme précédent pour qu'il affiche une liste de toutes les voitures correspondant au prix minimal.
11. Estimer le nombre de comparaisons effectuées (en fonction du nombre total n d'éléments dans la liste) dans le programme précédent (ou dans sa première version si vous n'avez pas réussi à faire la question précédente).

1 Troisième partie : Analyse numérique.

Notre concessionnaire désire désormais faire des tests sur une de ses voitures. Pour cela, il procède à l'expérience suivante : la voiture est lâchée à une vitesse initiale $v_0 > 0$, moteur coupé, et on observe l'évolution de sa vitesse au cours du temps (une expérience à ne pas tenter chez vous). On admet que la fonction $v(t)$ donnant cette évolution vérifie une équation différentielle du type $\frac{dv}{dt} = -\alpha - \beta v^2(t)$, où α et β sont des constantes strictement positives. On souhaite effectuer une résolution approchée à l'aide de la méthode d'Euler, et un gentil copain vous a déjà écrit la structure du programme (qui est tout de même légèrement incomplet) :

```
def Euler(f,v0,dt) :
    v,t,tmax = v0,0,92
    vitesse,temps=[v],[t]
    while t < tmax :
        ...
        ...
        ...
        ...
    return temps,vitesse
```

1. Expliquer le principe de la méthode d'Euler, et du programme ci-dessus. On précisera notamment clairement le rôle de toutes les variables apparaissant dans le programme. On expliquera précisément ce que représente f .
2. Compléter les lignes manquantes en milieu de programme (inutile de recopier le reste du programme).
3. Quel module Python permettrait de tracer une courbe à partir des deux listes retournées par la fonction **Euler**? Rappeler la commande permettant d'importer ce module en lui donnant un nom plus simple, et la commande permettant d'afficher le graphique.
4. Écrire la définition de la fonction f pour l'équation différentielle que nous cherchons à résoudre (on suppose que les constantes α et β ont déjà été définies au préalable).
5. On suppose que la fonction Euler nous a permis d'obtenir une liste **vitesse** contenant 500 valeurs successives de la vitesse de notre véhicule, et que cette liste de valeurs est strictement décroissante et devient négative à partir d'un certain point (ce n'est pas très réaliste, mais la modélisation choisie est cohérente avec cette constatation). On souhaite déterminer à quel endroit dans la liste apparaît la première valeur négative. Écrire un programme permettant d'obtenir cette valeur en parcourant tous les éléments de la liste jusqu'à en trouver un négatif (on utilisera un *while* pour cette question).
6. Proposer une méthode plus rapide pour déterminer cette valeur. Combien de tests faudra-t-il faire en moyenne avec cette nouvelle méthode pour trouver notre valeur? Comparer avec la première méthode utilisée.
7. Écrire une suite d'instructions permettant de trouver la valeur à l'aide de cette seconde méthode.



J'espère que vous n'avez pas trop eu besoin de moi ?