

Exercice : Gestion d'un inventaire de composants d'ordinateurs

Vous devez développer un programme Python permettant de gérer l'inventaire d'un magasin de composants d'ordinateurs. Chaque composant d'ordinateur est représenté par un dictionnaire contenant les informations suivantes :

- **id** : Identifiant unique du composant (entier)
- **nom** : Nom du composant (ex : "Processeur Intel i5", "Carte Graphique GTX 1050")
- **type** : Type du composant (ex : "Processeur", "Carte Graphique", "RAM", "Carte Mère")
- **quantité** : Quantité en stock (entier)
- **prix** : Prix du composant en euros (flottant)
- **date_ajout** : Date d'ajout en inventaire (format AAAA-MM-JJ)

Vous devez permettre les opérations suivantes sur la liste de composants :

- Ajouter, modifier, supprimer un composant.
- Rechercher et afficher des composants en fonction de divers critères.
- Trier et filtrer les composants par différents attributs.

Exercice : Gestion d'un inventaire de composants d'ordinateurs

Vous devez développer un programme Python permettant de gérer l'inventaire d'un magasin de composants d'ordinateurs. Chaque composant d'ordinateur est représenté par un dictionnaire contenant les informations suivantes :

- **id** : Identifiant unique du composant (entier)
- **nom** : Nom du composant (ex : "Processeur Intel i5", "Carte Graphique GTX 1050")
- **type** : Type du composant (ex : "Processeur", "Carte Graphique", "RAM", "Carte Mère")
- **quantité** : Quantité en stock (entier)
- **prix** : Prix du composant en euros (flottant)
- **date_ajout** : Date d'ajout en inventaire (format AAAA-MM-JJ)

Vous devez permettre les opérations suivantes sur la liste de composants :

- Ajouter, modifier, supprimer un composant.
- Rechercher et afficher des composants en fonction de divers critères.
- Trier et filtrer les composants par différents attributs.

Questions

1. Ajouter un composant

- a. Écrivez une fonction pour ajouter un nouveau composant à l'inventaire. La fonction doit prendre les informations du composant comme paramètres et générer un identifiant unique.

2. Modifier un composant

- a. Écrivez une fonction permettant de modifier les informations d'un composant donné. La fonction doit demander l'identifiant (`id`) du composant à modifier et permettre de mettre à jour ses informations (nom, type, quantité, prix, ou date d'ajout).

3. Supprimer un composant

- Écrivez une fonction pour supprimer un composant en fonction de son identifiant (`id`). La fonction doit afficher un message d'erreur si le composant n'existe pas.

4. Filtrage avancé par plusieurs critères

- Implémentez une fonction permettant de filtrer les composants selon plusieurs critères. Par exemple, affichez tous les composants de type "RAM" dont le prix est inférieur à 100 euros et dont la quantité en stock est supérieure à 10.

5. Filtrage par intervalle de prix

- Écrivez une fonction qui filtre les composants selon un intervalle de prix spécifié par l'utilisateur. Permettez aussi à l'utilisateur de spécifier plusieurs fourchettes de prix (par exemple, 50-100 euros ou 150-200 euros).

6. Recherche par mot-clé

- Créez une fonction de recherche permettant de trouver les composants contenant un mot-clé dans leur nom ou leur type (par exemple, "Intel" pour tous les composants contenant ce terme dans leur nom).

7. Tri combiné sur plusieurs critères

- Implémentez un tri basé sur deux critères. Par exemple :
 - Trier par type (ordre alphabétique) puis par prix (ordre croissant).
 - Trier par quantité (ordre décroissant) puis par nom (ordre alphabétique).
- Permettez à l'utilisateur de choisir l'ordre de tri (croissant ou décroissant) pour chaque critère.

8. Top des composants les plus chers ou les moins chers

- Écrivez une fonction pour afficher les `n` composants les plus chers ou les moins chers, où `n` est fourni par l'utilisateur.

9. Trouver les composants de même type mais de prix différents

- Créez une fonction qui affiche tous les composants du même type mais avec des prix différents. Par exemple, si plusieurs composants de type "RAM" ont des prix variés, affichez-les pour comparaison.

10. Rechercher le composant avec la quantité en stock la plus élevée

- Implémentez une fonction pour trouver et afficher le composant avec la quantité en stock la plus élevée. Modifiez ensuite cette fonction pour permettre d'afficher les n composants avec les quantités les plus élevées.

11. Composants ayant le même prix

- Écrivez une fonction pour afficher tous les composants ayant le même prix. Cette fonction permettra d'identifier des composants similaires en termes de coût.

12. Calculer la valeur totale des composants d'un type spécifique

- Écrivez une fonction qui calcule le prix total de tous les composants d'un type spécifique en multipliant le prix unitaire par la quantité en stock pour chaque composant. Demandez à l'utilisateur de saisir le type de composant pour effectuer le calcul.

13. Filtrer les composants par ancienneté dans l'inventaire

- En utilisant la clé `date_ajout`, implémentez une fonction permettant de filtrer les composants ajoutés avant ou après une date donnée.
- Ajoutez une option pour afficher les n composants les plus récents ou les plus anciens.