10_SDD

Dico changement d'indexage

Enoncé

On dispose d'un dictionnaire python qui indexe une liste d'étudiant par son année d'obtention du bac.
Un étudiant est représenté par un 5-uplets *(nom,prenom,numero,(j,m,annee),notes)* dont les éléments ont pour type respectif str, str, int, 3-uplets de int, liste de float.

notes est la liste des notes obtenues aux épreuves.
Pour simplifier, on considère que tous les étudiants ont une note pour toutes les épreuves.

Ecrire une fonction **transfo_indexage_num(dico_annee)** qui, étant donné un dictionnaire indexé par les années d'obtention du bac, construit et renvoie un dictionnaire indexé par le numéro étudiant. Le dictionnaire construit aura comme clés des numéros étudiants et les valeurs seront des 5-uplets représentant les étudiants.

Contraintes

La fonction doit s'appeler **transfo_indexage_num**.

Paramètres de la fonction

Un dictionnaire dont les clés sont des années d'obtention de bac, les valeurs sont des listes de 5-uplets (nom,prenom,numero,(j,m,annee),notes),

Retour de la fonction

Un dictionnaire dont les clés sont des numéros étudiants, les valeurs sont des 5-uplets (nom,prenom,numero, (j,m,annee),notes).

Exemples

Paramètres :

```
{
2016:
   [('nom2', 'prenom2', 40, (5, 7, 2016), [9.0, 11.0, 9.0]), ('nom5', 'prenom5', 100, (5, 7, 2
2017:
   [('nom4', 'prenom4', 80, (5, 7, 2017), [7.0, 13.0, 7.0]), ('nom7', 'prenom7', 140, (5, 7, 2
2015:
   [('nom3', 'prenom3', 60, (5, 7, 2015), [8.0, 12.0, 8.0]), ('nom6', 'prenom6', 120, (5, 7, 2
}
```

Retour :

```
{
160:('nom8', 'prenom8', 160, (5, 7, 2016), [3.0, 17.0, 3.0]),
80: ('nom4', 'prenom4', 80, (5, 7, 2017), [7.0, 13.0, 7.0]),
180: ('nom9', 'prenom9', 180, (5, 7, 2015), [2.0, 18.0, 2.0]),
100: ('nom5', 'prenom5', 100, (5, 7, 2016), [6.0, 14.0, 6.0]),
40: ('nom2', 'prenom2', 40, (5, 7, 2016), [9.0, 11.0, 9.0]),
140: ('nom7', 'prenom7', 140, (5, 7, 2017), [4.0, 16.0, 4.0]),
220: ('nom11', 'prenom11', 220, (5, 7, 2016), [10.0, 10.0, 10.0]),
200: ('nom10', 'prenom10', 200, (5, 7, 2017), [1.0, 19.0, 1.0]),
120: ('nom6', 'prenom6', 120, (5, 7, 2015), [5.0, 15.0, 5.0]),
60: ('nom3', 'prenom3', 60, (5, 7, 2015), [8.0, 12.0, 8.0])
}
```

Dico classement des moyennes par épreuve et par année

Enoncé

On dispose d'un dictionnaire python qui indexe une liste d'étudiant par son année d'obtention du bac.
Un étudiant est représenté par un 5-uplets *(nom,prenom,numero,(j,m,annee),notes)* dont les éléments ont pour type respectif str, str, int, 3-uplets de int, liste de float.

notes est la liste des notes obtenues aux épreuves.
Pour simplifier, on considère que tous les étudiants ont une note pour toutes les épreuves.

Ecrire une fonction **dico_classement_epreuves(dico)** qui, étant donné un dictionnaire indexé par les années d'obtention du bac, construit et renvoie un dictionnaire indexé par le numéro d'épreuve. Le dictionnaire construit aura comme clés les numéros d'épreuve et les valeurs seront des listes **triées** de couples (annee,moyenne). Les couples seront triés par ordre décroissant des moyennes.

Ainsi, dans le dictionnaire construit, pour chaque matière on obtiendra une liste qui classe les années de la meilleure à la moins bonne.

On propose d'utiliser le programme de tri à bulles dont voici une version pour trier en ordre croissant des entiers :

```
def tri_bulle(liste):
    # faire le travail n fois
    for i in range(len(liste)):
        # ne pas aller plus loin : les derniers sont tries
        for j in range(len(liste)-i-1):
            # permuter les voisins si besoin
            if liste[j]>liste[j+1] :
                liste[j],liste[j+1]=liste[j+1],liste[j]
    return liste
```

Vous pouvez faire appel à la fonction précédente **moyenne_annee_epreuve(dico, annee_du_bac, num_epreuve)** à l'aide du classique `from lib import moyenne_annee_epreuve`.

Contraintes

La fonction doit s'appeler **dico_classement_epreuves**.

Paramètres de la fonction

Un dictionnaire dont les clés sont des années d'obtention de bac, les valeurs sont des listes de 5-uplets (nom,prenom,numero,(j,m,annee),notes),

Retour de la fonction

Un dictionnaire dont les clés sont des numéros d'épreuves, les valeurs sont listes de couples (annee,moyenne). Les couples sont triés par ordre décroissant des moyennes.

Exemples

Paramètres :

```
{
2016:
   [('nom2', 'prenom2', 40, (5, 7, 2016), [9.0, 11.0, 9.0]), ('nom5', 'prenom5', 100, (5, 7, 2
2017:
   [('nom4', 'prenom4', 80, (5, 7, 2017), [7.0, 13.0, 7.0]), ('nom7', 'prenom7', 140, (5, 7, 2
2015:
   [('nom3', 'prenom3', 60, (5, 7, 2015), [8.0, 12.0, 8.0]), ('nom6', 'prenom6', 120, (5, 7, 2
}
```

Retour :

```
{
0: [(2016, 7.0), (2015, 5.0), (2017, 4.0)],
1: [(2017, 16.0), (2015, 15.0), (2016, 13.0)],
2: [(2016, 7.0), (2015, 5.0), (2017, 4.0)]
}
```

Dico Gestion étudiants - Création

Enoncé

On cherche à créer un dictionnaire python qui indexe une liste d'étudiant par son année d'obtention du bac.
Un étudiant est représenté par un tuple *(nom,prenom,numero,(j,m,annee),notes)* dont les éléments ont pour type respectif str, str, int, 3-uplets de int, liste de float.

notes est la liste des notes obtenues aux épreuves.
Pour simplifier, on considère que tous les étudiants ont une note pour toutes les épreuves.

Ecrire une fonction **liste_to_dico(liste_etudiants)** qui, étant donné une liste d'étudiants, construit et renvoie un tel dictionnaire.

Contraintes

La fonction doit s'appeler **liste_to_dico**.

Paramètres de la fonction

Une liste (pouvant être vide) de 5-uplets.

Retour de la fonction

Un dictionnaire.

Exemples

Paramètres :

```
[('nom2', 'prenom2', 40, (5, 7, 2016), [9.0, 11.0, 9.0]),('nom4', 'prenom4', 80, (5, 7, 2017),
 [7.0, 13.0, 7.0]),('nom3', 'prenom3', 60, (5, 7, 2015), [8.0, 12.0, 8.0]),('nom1', 'prenom1', 2
```

Retour :

```
  {
  2016: [('nom2', 'prenom2', 40, (5, 7, 2016), [9.0, 11.0, 9.0])],
  2017: [('nom4', 'prenom4', 80, (5, 7, 2017), [7.0, 13.0, 7.0])],
  2015: [('nom3', 'prenom3', 60, (5, 7, 2015), [8.0, 12.0, 8.0]),('nom1', 'prenom1', 20, (5, 7,
  }
```

Attention à l'affichage : un `print(dico)` ne donnera pas ce résultat. Ici on demande un retour de fonction, pas un affichage.

Dico indexé par année - Moyenne

Enoncé

On dispose d'un dictionnaire python qui indexe une liste d'étudiant par son année d'obtention du bac.
Un étudiant est représenté par un 5-uplets *(nom,prenom,numero,(j,m,annee),notes)* dont les éléments ont pour type respectif str, str, int, 3-uplets de int, liste de float.

notes est la liste des notes obtenues aux épreuves.
Pour simplifier, on considère que tous les étudiants ont une note pour toutes les épreuves.

Ecrire une fonction **moyenne_annee_epreuve(dico,annee,epreuve)** qui, étant donné un dictionnaire, une année d'obtention de bac et un numéro d'épreuve, calcule et renvoie la moyenne à cette épreuve pour les étudiants ayant obtenu leur bac à l'année demandée. Si le dictionnaire est vide ou si le numéro d'épreuve n'existe pas alors la fonction renvoie 0.0.

Contraintes

La fonction doit s'appeler **moyenne_annee_epreuve**.

Paramètres de la fonction

- Un dictionnaire dont les clés sont des années d'obtention de bac, les valeurs sont des listes de 5-uplets (nom,prenom,numero,(j,m,annee),notes),
- une année du bac (int),
- un numéro d'épreuve (int).

Retour de la fonction

Un float.

Exemples

Paramètres :

```
{
  2016:
     [('nom2', 'prenom2', 40, (5, 7, 2016), [9.0, 11.0, 9.0])],
  2017:
     [('nom4', 'prenom4', 80, (5, 7, 2017), [7.0, 13.0, 7.0]),('nom6', 'prenom6', 76, (5, 7, 201
  2015:
  [('nom3', 'prenom3', 60, (5, 7, 2015), [8.0, 12.0, 8.0])]
}
2017
0
```

Retour :

```
8.0
```

Dico indexé par numéros - Moyenne

Enoncé

On dispose d'un dictionnaire python qui indexe des étudiants par leur numéro.
Un étudiant est représenté par un 5-uplets *(nom,prenom,numero,(j,m,annee),notes)* dont les éléments ont pour type respectif str, str, int, 3-uplets de int, liste de float.

notes est la liste des notes obtenues aux épreuves.
Pour simplifier, on considère que tous les étudiants ont une note pour toutes les épreuves.

Ecrire une fonction **moyenne_num_annee_epreuve(dico,annee,epreuve)** qui, étant donné un dictionnaire, une année d'obtention de bac et un numéro d'épreuve, calcule et renvoie la moyenne à cette épreuve pour les étudiants ayant obtenu leur bac à l'année demandée. Si le dictionnaire est vide ou si le numéro d'épreuve n'existe pas alors la fonction renvoie 0.0.

Contraintes

La fonction doit s'appeler **moyenne_num_annee_epreuve**.

Paramètres de la fonction

Un dictionnaire dont les clés sont des numéros d'étudiant, les valeurs sont des 5-uplets (nom,prenom,numero, (j,m,annee),notes),

Retour de la fonction

Un float.

Exemples

Paramètres :

```
{
  40: ('nom2', 'prenom2', 40, (5, 7, 2016), [9.0, 11.0, 9.0]),
  80: ('nom4', 'prenom4', 80, (5, 7, 2017), [7.0, 13.0, 7.0]),
  60: ('nom3', 'prenom3', 60, (5, 7, 2015), [8.0, 12.0, 8.0])}
2017
0
```

Retour :

```
7.0
```

Création dictionnaire taille mot

Sujet

Soit une liste de mots (trouvés dans un texte par exemple), on vous demande de construire le dictionnaire qui associé à une taille de mot l'ensemble des mots de cette taille. Exemple :

```
["bonjour", "toto", "beau", "maison", "joli", "taille", "de", "le", "le"]
```

Doit renvoyer le dictionnaire suivant :

```
{7: {'bonjour'}, 4: {'toto', 'beau', 'joli'}, 6: {'taille', 'maison'}, 2: {'le', 'de'}}
```

En effet, le mot `'bonjour'` contient 7 lettres (et c'est le seul de la liste) : la clé 7 est associée à l'ensemble ne contenant que le mot `'bonjour'`.
Les trois mots `'toto'`, `'beau'` et `'joli'` sont tous de taille 4 : la clé 4 est associée est à l'ensemble contenant ces trois mots. Le mot `'le'` est présent deux fois dans la liste, mais n'apparait qu'une fois dans l'ensemble associé à la clé 2.

Entrée

Une liste de chaînes de caractères.

Sortie

Un dictionnaire dont les clés sont des entiers et les valeurs sont des ensembles de chaînes de caractères.

Exemples

Entrée :

```
["mot1", "mot2", "abc","abc","ab","c", "d", "abc", "mot1"]
```

Sortie :

```
{4: {'mot2', 'mot1'}, 3: {'abc'}, 2: {'ab'}, 1: {'c', 'd'}}
```

Entrée :

```
[]
```

Sortie :

```
{}
```