|  |  |  |
| --- | --- | --- |
| Spécialité NSI Première  | **DEVOIR SURVEILLE DE** | Vendredi 4 février 2022 |
| Lycée d’Avesnières | **NSI** | Durée : 55 mn |
| Année scolaire 2021-2022 | **N° 5** | Calculatrice interdite |

**Exercice 1** (14 points)

**Partie 1**

On dispose d'un fichier csv nommé capitales.csv dont voici les 20 première lignes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| nom\_pays | nom\_capitale | latitude | longitude | code\_pays | continent |
| Somaliland | Hargeisa | 9.55 | 44.050000 | NULL | Africa |
| South Georgia and South Sandwich Islands | King Edward Point | -54.283333 | -36.500000 | GS | Antarctica |
| French Southern and Antarctic Lands | Port-aux-Français | -49.35 | 70.216667 | TF | Antarctica |
| Palestine | Jerusalem | 31.766666666666666 | 35.233333 | PS | Asia |
| Aland Islands | Mariehamn | 60.116667 | 19.900000 | AX | Europe |
| Nauru | Yaren | -0.5477 | 166.920867 | NR | Australia |
| Saint Martin | Marigot | 18.0731 | -63.082200 | MF | North America |
| Tokelau | Atafu | -9.166667 | -171.833333 | TK | Australia |
| Western Sahara | El-Aaiún | 27.153611 | -13.203333 | EH | Africa |
| Afghanistan | Kabul | 34.516666666666666 | 69.183333 | AF | Asia |
| Albania | Tirana | 41.31666666666667 | 19.816667 | AL | Europe |
| Algeria | Algiers | 36.75 | 3.050000 | DZ | Africa |
| American Samoa | Pago Pago | -14.266666666666667 | -170.700000 | AS | Australia |
| Andorra | Andorra la Vella | 42.5 | 1.516667 | AD | Europe |
| Angola | Luanda | -8.833333333333334 | 13.216667 | AO | Africa |
| Anguilla | The Valley | 18.216666666666665 | -63.050000 | AI | North America |
| Antigua and Barbuda | Saint John's | 17.116666666666667 | -61.850000 | AG | North America |
| Argentina | Buenos Aires | -34.583333333333336 | -58.666667 | AR | South America |
| Armenia | Yerevan | 40.166666666666664 | 44.500000 | AM | Europe |
| Aruba | Oranjestad | 12.516666666666667 | -70.033333 | AW | North America |
| Aland Islands | Mariehamn | 60.11666 | 19.90000 | AX | Europe |

On dispose également d'un fichier Python exercice.py placé dans le même répertoire que le fichier capitales.csv :

import csv

#ouverture du fichier csv

def lecture\_fichier(nom\_fichier):

    with open(nom\_fichier, mode='r', encoding='utf-8-sig') as fichier\_ouvert:

        return [ligne for ligne in csv.reader(fichier\_ouvert)]

1. Quels sont les descripteurs de la table ?
2. On souhaite récupérer le contenu du fichier csv dans le programme Python .

Quelle instruction Python faut-il écrire afin de lire le fichier capitales.csv et d'affecter son contenu dans la variable appelée table\_capitales ?

1. Quel est le type de la variable table\_capitales ?
2. On ajoute la fonction selectionner(table, critere) au programme Python :

def selectionner(table, critere):

    selection=[]

    for i in range(1,len(table)):

        if critere(table[i]):

            selection.append(table[i])

    return selection

On exécute la ligne :

recherche\_1 = selectionner(table\_capitales, lambda x: float(x[2]) < 0)

Que va contenir la variable recherche\_1 ? Expliquez votre réponse.

1. Grâce à la fonction selectionner(table, critere) du programme, donnez le critère de sélection à écrire en argument pour renvoyer uniquement les capitales situées sur le continent sud-américain et affecter le résultat de la recherche dans une variable nommée recherche\_2.
2. Grâce à la fonction selectionner(table, critere) du programme, donnez le critère de sélection à écrire en argument pour renvoyer uniquement les capitales d'Amérique du sud située dans l'hémisphère Nord dans une variable nommée recherche\_3.

**Partie 2**

On dispose d'un fichier csv nommé sacs.csv dans le même répertoire que le fichier exercice.py.

On a importé dans le programme Python le fichier sacs.csv en exécutant l'instruction

table\_sacs = lecture\_fichier('sacs.csv').

Cela a permis d'affecter à la variable table\_sacs la valeur :

[['id', 'nom', 'longueur', 'largeur', 'hauteur', 'couleur', 'poids', 'prix\_HT', 'fabrique en France'], ['1', 'trousse', '20', '6', '7', 'bleu', '150', '18', 'oui'], ['2', 'polochon', '50', '25', '25', 'rouge', '450', '79', 'oui'], ['3', 'cabas', '35', '14', '30', 'bleu', '330', '95', 'non'], ['4', 'besace', '24', '21', '9', 'marron', '500', '84', 'oui'], ['5', 'banane', '23', '9', '13', 'noir', '260', '28', 'oui'], ['6', 'baluchon', '61', '38', '22', 'beige', '900', '145', 'non'], ['7', 'sac à dos', '30', '40', '18', 'gris', '375', '50', 'oui'], ['8', 'seau', '24', '25', '13.5', 'noir', '430', '110', 'oui'], ['9', 'wrist bag', '27', '16.5', '3.5', 'bleu', '125', '28', 'non'], ['10', 'minaudiere', '23', '5', '8', 'noir', '160', '35', 'oui']]

1. Écrire une instruction Python qui permet d'affecter à une variable nommée les\_descripteurs la liste des descripteurs de la table table\_sacs.
2. Il est possible de trier la table grâce à la fonction sorted() intégrée à Python et à l'utilisation d'une fonction lambda. Que va contenir la liste de listes s1 après l'instruction suivante ?

s1 = sorted(table\_sacs[1:], key=lambda ligne: float(ligne[2]))

On ne demande pas d'écrire la liste de listes s1 complètement, mais de décrire ce qu'elle contient.

1. Même question avec

s2 = sorted(table\_sacs[1:], key=lambda ligne: float(ligne[7]), reverse=True)

1. Écrire une instruction Python qui permet d'affecter à une variable nommée ma\_selection les cinq sacs les plus chers.
2. Soit la variable s3 la liste de listes table\_sacs rangée par ordre alphabétique des noms de sacs.
	1. Écrire l'instruction Python permettant d'obtenir s3.
	2. Écrire l'instruction Python permettant d'obtenir à partir de la table triée par noms s3 uniquement les sacs de couleur noire.

**Exercice 2** (6 points)

On souhaite concaténer les deux tables ci-contre qui contiennent les données recueillies lors d'une expérience scientifique franco-américaine.

T désigne une température et tps une durée en seconde. La difficulté est que les températures T dans les mesures françaises sont en degrés Celsius et que les températures T dans les mesures américaines sont en degrés Fahrenheit. Mais on connait la formule qui permet de transformer une température Fahrenheit $T\_{F}$ en une température Celsius $T\_{C}$ :

$$T\_{C}= \left(T\_{F}-32\right)×\frac{5}{9} $$

On a créé deux listes de listes nommées table1 et table2 contenant les informations des deux tables décrites ci-dessus.

table1 = [['id\_exp', 'T', 'tps'], [1, 20.2, 56], [2, 15.5, 85], [3, 18.6, 120]]

table2 = [['id\_exp', 'T', 'tps'], [1, 53.6, 210], [2, 62.6, 81], [3, 66.2, 70]]

1. Les deux tables ont-elles les mêmes descripteurs ?
2. Betty a écrit la fonction betty(t1, t2) suivante :

def betty(t1, t2):

    """

    Réalise la concaténation de t1 et t2.

    Paramètres :

    ------------

            t1 et t2 de type liste de listes.

    Retourne :

    ----------

            t1 + t2[1:] de type liste de listes.

    """

    return t1 + t2[1:]

puis elle essaye sa fonction :

resultat\_betty = betty(table1, table2)

Elle obtient la table resultat\_betty qui vaut :

[['id\_exp', 'T', 'tps'], [1, 20.2, 56], [2, 15.5, 85], [3, 18.6, 120], [1, 53.6, 210], [2, 62.6, 81], [3, 66.2, 70]]

Est-ce que cette façon de procéder est satisfaisante ? Expliquez.

1. John veut améliorer la fonction de Betty et a écrit la fonction fusion(t1, t2) de la façon suivante :

def fusion(t1, t2):

    """

    Réalise la conversion des températures de t2 de degrés F en degrés C.

    Numérote à partir de 4 les id\_exp de t2.

    Concatène t1 et t2.

    Paramètres :

    ------------

            t1 et t2 de type liste de listes.

    Retourne :

    ----------

            t1 + t2[1:] de type liste de listes.

    """

    #  Conversion des températures de t2 en coupant sa ligne des descripteurs.

    for ligne in t2[1:]:

        ...

    #  Renumérote les id\_exp de t2 en coupant sa ligne des descripteurs.

    for ligne in t2[1:]:

        ...

    return t1 + t2[1:]

puis il essaye sa fonction : resultat\_john = fusion(table1, table2)

Sur votre copie, écrivez **uniquement les deux boucles for** de façon à obtenir resultat\_john qui vaut :

[['id\_exp', 'T', 'tps'], [1, 20.2, 56], [2, 15.5, 85], [3, 18.6, 120], [4, 12.0, 210], [5, 17.0, 81], [6, 19.0, 70]]

Remarque : Améliorez le résultat de la conversion en degrés Celsius en arrondissant à 1 décimale à l'aide de la fonction intégrée dans Python round(a, n) qui arrondit le nombre a à $10^{-n}$ près.