

# BAC 2021 : AMÉRIQUE, OCÉANIE, ASIE, LIBAN (CENTRES ÉTRANGERS DU GROUPE 2)

## Exercice 3

Cet exercice porte sur les tableaux et sur la programmation de base en Python.

On rappelle que `len` est une fonction qui prend un tableau en paramètre et renvoie sa longueur.

C'est-à-dire le nombre d'éléments présents dans le tableau.

Exemple : `len([12, 54, 34, 57])` vaut 4.

Le but de cet exercice est de programmer différentes réductions pour un site de vente de vêtements en ligne.

On rappelle que si le prix d'un article avant réduction est de  $x$  euros,

— son prix vaut  $0,5x$  si on lui applique une réduction de 50%,

— son prix vaut  $0,6x$  si on lui applique une réduction de 40%,

— son prix vaut  $0,7x$  si on lui applique une réduction de 30%,

— son prix vaut  $0,8x$  si on lui applique une réduction de 20%,

— son prix vaut  $0,9x$  si on lui applique une réduction de 10%.

Dans le système informatique du site de vente, l'ensemble des articles qu'un client veut acheter, appelé panier, est modélisé par un tableau de flottants.

Par exemple, si un client veut acheter un pantalon à 30,50 euros, un tee-shirt à 15 euros, une paire de chaussettes à 6 euros, une jupe à 20 euros, une paire de collants à 5 euros, une robe à 35 euros et un short à 10,50 euros, le système informatique aura le tableau suivant :

```
tab = [30.5, 15.0, 6.0, 20.0, 5.0, 35.0, 10.5].
```

1. (a) Écrire une fonction Python `total_hors_reduction` ayant pour argument le tableau des prix des articles du panier d'un client et renvoyant le total des prix de ces articles.

(b) Le site de vente propose la promotion suivante comme offre de bienvenue : 20% de réduction sur le premier article de la liste, 30% de réduction sur le deuxième article de la liste (s'il y a au moins deux articles) et aucune réduction sur le reste des articles (s'il y en a).

Recopier sur la copie et compléter la fonction Python `offre_bienvenue` prenant en paramètre le tableau `tab` des prix des articles du panier d'un client et renvoyant le total à payer lorsqu'on leur applique l'offre de bienvenue.

```
1 def offre_bienvenue ( tab ) :
2     """ tableau -> float """
3     somme = 0
4     longueur = len ( tab )
5     if longueur > 0 :
6         somme = tab [0]*...
7     if longueur > 1 :
8         somme = somme + ...
9     if longueur > 2 :
10        for i in range (2 , longueur ) :
11            somme =...
12        return ...
```

Pour toute la suite de l'exercice, on pourra utiliser la fonction `total_hors_reduction` même si la question 1 n'a pas été traitée.

21-NSIJ1AN1 Page : 6/12

2. Lors de la période des soldes, le site de vente propose les réductions suivantes :

— si le panier contient 5 articles ou plus, une réduction globale de 50%,

— si le panier contient 4 articles, une réduction globale de 40%,

— si le panier contient 3 articles, une réduction globale de 30%,

— si le panier contient 2 articles, une réduction globale de 20%,

— si le panier contient 1 article, une réduction globale de 10%.

Proposer une fonction Python `prix_solde` ayant pour argument le tableau `tab` des prix des articles du panier d'un client et renvoyant le total des prix de ces articles lorsqu'on leur applique la réduction des soldes.

3. (a) Écrire une fonction `minimum` qui prend en paramètre un tableau `tab` de nombres et renvoie

la valeur minimum présente dans le tableau.

(b) Pour ses bons clients, le site de vente propose une offre promotionnelle, à partir de 2 articles achetés, l'article le moins cher des articles commandés est offert.

Écrire une fonction Python `offre_bon_client` ayant pour paramètre le tableau des prix des articles du panier d'un client et renvoyant le total à payer lorsqu'on leur applique l'offre bon client.

4. Afin de diminuer le stock de ses articles dans ses entrepôts, l'entreprise imagine faire l'offre suivante à ses clients : en suivant l'ordre des articles dans le panier du client, elle considère les 3 premiers articles et offre le moins cher, puis les 3 suivants et offre le moins cher et ainsi de suite jusqu'à ce qu'il reste au plus 2 articles qui n'ont alors droit à aucune réduction.

Exemple : Si le panier du client contient un pantalon à 30,50 euros, un tee-shirt à 15 euros, une paire de chaussettes à 6 euros, une jupe à 20 euros, une paire de collants à 5 euros, une robe à 35 euros et un short à 10,50 euros, ce panier est représenté par le tableau suivant :

```
tab = [30.5, 15.0, 6.0, 20.0, 5.0, 35.0, 10.5]
```

Pour le premier groupe (le pantalon à 30,50 euros, le tee-shirt à 15 euros, la paire de chaussettes à 6 euros), l'article le moins cher, la paire de chaussettes à 6 euros, est offert. Pour le second groupe (la jupe à 20 euros, la paire de collants à 5 euros, la robe à 35 euros), la paire de collants à 5 euros est offerte.

Donc le total après promotion de déstockage est 111 euros.

On constate que le prix après promotion de déstockage dépend de l'ordre dans lequel se présentent les articles dans le panier.

(a) Proposer un panier contenant les mêmes articles que ceux de l'exemple mais ayant un prix après promotion de déstockage différent de 111 euros.

(b) Proposer un panier contenant les mêmes articles mais ayant le prix après promotion de déstockage le plus bas possible.

(c) Une fois ses articles choisis, quel algorithme le client peut-il utiliser pour modifier son panier afin de s'assurer qu'il obtiendra le prix après promotion de déstockage le plus bas possible ? On ne demande pas d'écrire cet algorithme.

## BAC 2021 : CENTRES ÉTRANGERS DU GROUPE 1

### Exercice 2

Notion abordée : structures de données (dictionnaires)

Une ville souhaite gérer son parc de vélos en location partagée. L'ensemble de la flotte de vélos est stocké dans une table de données représentée en langage Python par un dictionnaire contenant des associations de type `id_velo : dict_velo` où `id_velo` est un nombre entier compris entre 1 et 199 qui correspond à l'identifiant unique du vélo et `dict_velo` est un dictionnaire dont les clés sont : "type", "etat", "station".

Les valeurs associées aux clés "type", "etat", "station" de `dict_velo` sont de type chaînes de caractères ou nombre entier :

- "type" : chaîne de caractères qui peut prendre la valeur "electrique" ou "classique"
- "etat" : nombre entier qui peut prendre la valeur 1 si le vélo est disponible, 0 si le vélo est en déplacement, -1 si le vélo est en panne
- "station" : chaînes de caractères qui identifie la station où est garé le vélo.

Dans le cas où le vélo est en déplacement ou en panne, "station" correspond à celle où il a été dernièrement stationné.

Voici un extrait de la table de données :

```
flotte = {
12 : {"type" : "electrique", "etat" : 1, "station" : "Prefecture"},
80 : {"type" : "classique", "etat" : 0, "station" : "Saint-Leu"},
45 : {"type" : "classique", "etat" : 1, "station" : "Baraban"},
41 : {"type" : "classique", "etat" : -1, "station" : "Citadelle"},
26 : {"type" : "classique", "etat" : 1, "station" : "Coliseum"},
28 : {"type" : "electrique", "etat" : 0, "station" : "Coliseum"},
74 : {"type" : "electrique", "etat" : 1, "station" : "Jacobins"},
13 : {"type" : "classique", "etat" : 0, "station" : "Citadelle"},
```

```
83 : {"type" : "classique", "etat" : -1, "station" : "Saint-Leu"},
22 : {"type" : "electrique", "etat" : -1, "station" : "Joffre"}
}
```

`flotte` étant une variable globale du programme.

Toutes les questions de cet exercice se réfèrent à l'extrait de la table `flotte` fourni ci-dessus.

L'annexe 1 présente un rappel sur les dictionnaires en langage Python.

1.

1.a. Que renvoie l'instruction `flotte[26]` ?

1.b. Que renvoie l'instruction `flotte[80]["etat"]` ?

1.c. Que renvoie l'instruction `flotte[99]["etat"]` ?

2. Voici le script d'une fonction :

```
def proposition(choix):
    for v in flotte:
        if flotte[v]["type"] == choix and flotte[v]["etat"] == 1:
            return flotte[v]["station"]
```

2.a. Quelles sont les valeurs possibles de la variable `choix` ?

2.b. Expliquer ce que renvoie la fonction lorsque l'on choisit comme paramètre l'une des valeurs possibles de la variable `choix`.

3.

3.a. Écrire un script en langage Python qui affiche les identifiants (`id_velo`) de tous les vélos disponibles à la station "Citadelle".

3.b. Écrire un script en langage Python qui permet d'afficher l'identifiant (`id_velo`) et la station de tous les vélos électriques qui ne sont pas en panne.

4. On dispose d'une table de données des positions GPS de toutes les stations, dont un extrait est donné ci-dessous. Cette table est stockée sous forme d'un dictionnaire.

Chaque élément du dictionnaire est du type : 'nom de la station' : (latitude, longitude)

```
stations = {
    'Prefecture' : (49.8905, 2.2967) ,
    'Saint-Leu' : (49.8982, 2.3017),
    'Coliseum' : (49.8942, 2.2874),
    'Jacobins' : (49.8912, 2.3016)
}
```

On admet que l'on dispose d'une fonction `distance(p1, p2)` permettant de renvoyer la distance en mètres entre deux positions données par leurs coordonnées GPS (latitude et longitude).

Cette fonction prend en paramètre deux tuples représentant les coordonnées des deux positions GPS et renvoie un nombre entier représentant cette distance en mètres.

Par exemple, `distance((49.8905, 2.2967), (49.8912, 2.3016))` renvoie 9591

Écrire une fonction qui prend en paramètre les coordonnées GPS de l'utilisateur sous forme d'un tuple et qui renvoie, pour chaque station située à moins de 800 mètres de l'utilisateur :

- le nom de la station ;
- la distance entre l'utilisateur et la station ;
- les identifiants des vélos disponibles dans cette station.

Une station où aucun vélo n'est disponible ne doit pas être affichée.

**Annexe 1 (exercice 2)**  
**(à ne pas rendre avec la copie)**

Action	Instruction et syntaxe
Créer un dictionnaire vide	<code>dico={}</code>
Obtenir un élément d'un dictionnaire existant à partir de sa clé renvoie une erreur si <code>cle</code> n'existe pas dans le dictionnaire	<code>dico[cle]</code>
Modifier la valeur d'un élément d'un dictionnaire à partir de sa clé	<code>dico[cle]=nouvelle_valeur</code>
Ajouter un élément dans un dictionnaire existant	<code>dico[nouvelle_cle]=valeur</code>
Supprimer et obtenir un élément d'un dictionnaire à partir de sa clé	<code>dico.pop(cle)</code>
Tester l'appartenance d'un élément à un dictionnaire (renvoie un booléen)	<code>cle in dico</code>
Objet itérable contenant les clés. 🐁 ce n'est pas un objet de type list	<code>dico.keys()</code>
Objet itérable contenant les valeurs. 🐁 ce n'est pas un objet de type list	<code>dico.values()</code>
Objet itérable contenant les couples (clé,valeur)	<code>dico.items()</code>
Afficher les associations <code>cle:valeur</code> du dictionnaire <code>dico</code>	<pre>for cle in dico:     print(cle, dico[cle])</pre>