

Exercice 3 (4 points).

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.

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.