DS n° 04 – Initiation à l'algorithmie

- Faire tous les exercices dans un fichier NomPrenom.py à sauvegarder,
- mettre en commentaire l'exercice et la question traités (ex : # Exercice 1),
- ne pas oublier pas de commenter ce qui est fait dans votre code (ex : # Je crée une fonction pour calculer la racine d'un nombre),
- les deux parties sont indépendantes et peuvent être traitées dans l'ordre que vous voulez,
- il faut vérifier avant de partir que le code peut s'exécuter et qu'il affiche les résultats que vous attendez. Les lignes de code qui doivent s'exécuter sont décommentées.

Introduction

Un supermarché met en place un système de passage automatique en caisse. Un client scanne les articles à l'aide d'un scanner de code-barres au fur et à mesure qu'il les ajoute dans son panier.

Les articles s'enregistrent alors dans une structure de données.

Le panier d'un client sera représenté par une file contenant les articles scannés.

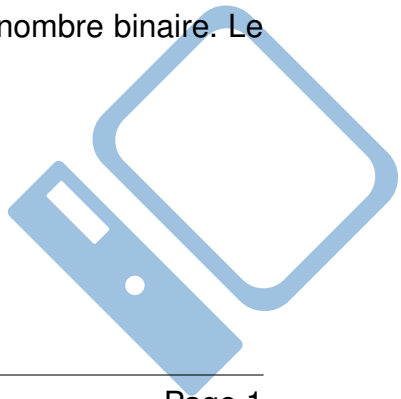
Les articles sont représentés par des dictionnaires `(code_barre, designation, prix, horaire_scan)` où :

- `code_barre` est un nombre entier identifiant l'article,
- `designation` est une chaîne de caractères qui pourra être affichée sur le ticket de caisse,
- `prix` est un nombre décimal donnant le prix d'une unité de cet article,
- `horaire_scan` est un nombre entier de secondes permettant de connaître l'heure où l'article a été scanné

1 Conversion décimal-binaire

Question 1 **Coder** une fonction `inverse_liste(liste)` qui prend en entrée une liste, et retourne la liste dans l'ordre inverse. L'utilisation de la fonction `reverse()` n'est pas autorisée. **Afficher** le résultat de la commande `inverse_liste(['A','B','C'])`

La fonction `convert_bin(decimal)` convertit un nombre décimal en nombre binaire. Le résultat est affiché sous la forme d'une **liste** de 0 et de 1.



```
def convert_bin(decimal):
    binaire=[]
    r=decimal
    while r>1:
        .....
        .....
        binaire.append(r)
        r=q
    binaire.append(r)
    return ........
```

Question 2 **Compléter** la fonction `convert_bin(decimal)`. La commande `convert_bin(19)` renvoie `[1,0,0,1,1]`.

Si la question précédente n'est pas traitée, le code suivant pourra être utilisé, il renvoie le résultat attendu.

```
def convert_bin(decimal):
    a=str(bin(decimal))[2:]
    s=[*a]
    s=[int(i) for i in s]
    return s
```

On rappelle que sous forme polynomiale, un nombre binaire quelconque est exprimé par : $N = \sum_{0}^{n} \alpha_i.2^i$, avec $\alpha_i = 0$ ou 1.

Question 3 **Coder** une fonction `convert_decimal(binaire)` qui convertit un nombre binaire sous la forme d'une liste en décimal. **Afficher** le résultat de `convert_decimal(convert_bin(19))` qui doit renvoyer 19.

2 Manipulation du panier

Le premier `produit` scanné possède les caractéristiques suivantes :

— 'code barre' : 31002,

— 'designation' :"café noir",

— 'prix' :1.50,

— 'horaire_scan' :50525.

Question 4 **Stocker** dans un dictionnaire `produit` les informations précédentes. **Afficher** `produit`.

La liste `panier` contiendra l'ensemble des produits scannés, afin d'être achetés, par la personne dans le magasin.

Question 5 **Coder** une fonction `enfiler(panier,produit)` qui enfile un nouveau `produit` dans le `panier`. Après avoir exécuté la commande `enfiler(panier,produit)`, **afficher** le `panier`, il doit contenir le `produit` précédemment créé.

On considère maintenant que les clients peuvent faire les courses à plusieurs et qu'il est possible d'ajouter un autre panier dans notre panier à tout moment.

Soit le panier 2 suivant :

- `panier[0]` :
  - 'code barre' :20333,
  - 'designation' :"madeleines",
  - 'prix' :4.30,
  - 'horaire_scan' :40520.
- `panier[1]` :
  - 'code barre' :1204,
  - 'designation' :"sucre",
  - 'prix' :3.40,
  - 'horaire_scan' :40550.
- `panier[2]` :
  - 'code barre' :2046,
  - 'designation' :"chocolat",
  - 'prix' :2.50,
  - 'horaire_scan' :40612.

Question 6 **Stocker** dans une liste l'ensemble des dictionnaires correspondant au panier 2 précédent. **Afficher** `panier2`.

Question 7 **Coder** une fonction `remplir(panier,panier2)` qui enfile tout le contenu de `panier 2` dans `panier`. Après avoir exécuté la commande `remplir(panier,panier2)`, **afficher** le contenu de `panier`.

Au fur et à mesure, il est nécessaire de connaître le coût total du panier.

Question 8 **Coder** la fonction `prix_total(panier)` qui renvoie le coût total du panier. **Afficher** le résultat de la commande `prix_total(panier)`.

A des fins de statistiques le magasin a besoin de connaître la durée de la présence des clients dans le magasin. Une méthode de mesure consiste à mesure la durée écoulée entre la mise dans le panier du premier achat et du dernier.

Question 9 **Coder** une fonction `duree_achats(panier)` qui renvoie la durée écoulée entre le premier achat effectué et le dernier. Attention, à cause de l'import d'autres paniers, les produits ne sont pas forcément stocké dans la liste dans leur ordre d'achat. **Afficher** le résultat de la commande `duree_achats(panier)`.

Question 10 Le résultat, en secondes, étant peu lisible, **coder** une fonction `convertion_duree(duree)` qui prend en entrée une `duree` en seconde et qui retourne le résultats sous le format suivant :

```
return "{} heures, {} minutes et {} secondes".format(heures,
minutes, secondes)
```

où `heures`, `minutes`, `secondes` correspondent à la conversion de `duree`. **Afficher** la durée des achats du panier en cours sous une forme lisible.

A la caisse il est possible de transformer ses courses en commande internet. Cette fonction est réalisée par le code suivant :

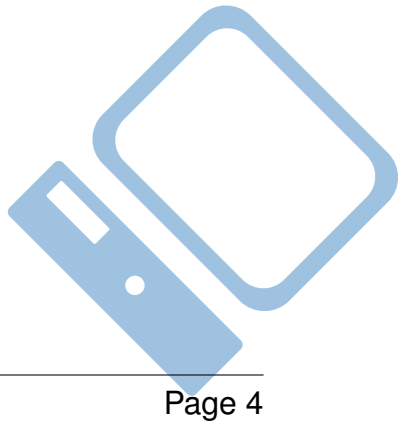
```
produits_commandes=[]
def commander(produit):
    produits_commandes.append(produit)

def defiler(panier):
    .....
    .....

def commander_panier(panier):
    while len(panier)>0:
        defiler(panier)
```

Question 11 **Recopier** ce code et **compléter** la fonction `defiler(panier)`. A l'issue de l'exécution de ce code, le contenu de `panier` doit être vidé et intégralement transféré dans `produits_commandes`. Afin de vérifier ce résultat, **afficher** `panier` avant l'exécution de `commander_panier(panier)`, puis afficher de nouveau `panier` et `produits_commandes`.

FIN



Question 1

```python
# -*- coding: utf-8 -*-
# Question 1
def inverse_liste(liste):
    return [liste[i] for i in range(len(liste)-1,-1,-1)]

print('Question 1')
print(inverse_liste(['A','B','C']))
```

Question 2

```python
# Question 2
def convert_bin(decimal):
    binaire=[]
    r=decimal
    while r>1:
        q=r//2
        r=r%2
        binaire.append(r)
        r=q
    binaire.append(r)
    return inverse_liste(binaire)

def convert_bin_default(decimal):
    a=str(bin(decimal))[2:]
    s=[*a]
    s=[int(i) for i in s]
    return s

print('\nQuestion 2')
print(convert_bin(19))
print(convert_bin_default(19))
```

Question 3

```python
# Question 3
def convert_decimal(binaire):
    decimal=0
    for i in range(len(binaire)):
        decimal+=binaire[len(binaire)-1-i]*2**i
    return decimal

print('\nQuestion 3')
print(convert_decimal(convert_bin(19)))
```

Question 4

```python
#Question 4
produit={'code barre':31002,'designation':"café noir",'prix':1.50,'horaire_scan':50525}
```

print(produit)

Question 5

```python
#Question 5
panier=[]
def enfiler(panier,produit):
    # Méthode qui enfile le produit au panier
    panier.append(produit)

enfiler(panier,produit)
print('\nQuestion 5')
print(panier)
```

Question 6

```python
#Question 6
panier2=[{'code barre':20333,'designation':"madeleines",'prix':4.30,'horaire_scan':40520
                    {'code barre':1204,'designation':"sucre",'prix':3.40,'horaire_scan':4
                    {'code barre':2046,'designation':"chocolat",'prix':2.50,'horaire_scan
```

print(panier2)

Question 7

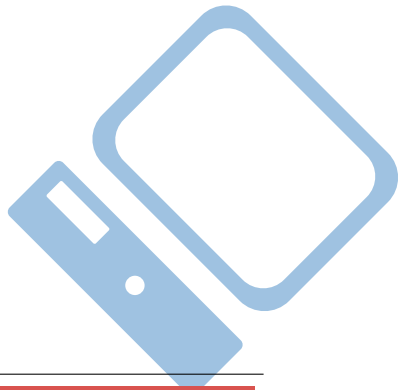
```python
#Question 7
def remplir(panier,panier2):
    # Méthode qui enfile un panier 2 dans un panier
    for element in panier2:
        panier.append(element)

remplir(panier,panier2)
print('\nQuestion 7')
print(panier)
```

Question 8

```python
#Question 8
def prix_total(panier):
    prix=0
    for element in panier:
        prix+=element['prix']
    return prix

print('\nQuestion 8')
print(prix_total(panier))
```



Question 9

```python
#Question 9
def duree_achats(panier):
    min=panier[0]['horaire_scan']
    max=panier[0]['horaire_scan']
    for element in panier[1:]:
        if element['horaire_scan']<min:
            min=element['horaire_scan']
        if element['horaire_scan']>max:
            max=element['horaire_scan']
    return max-min

print('\nQuestion 9')
print(duree_achats(panier))
```

Question 10

```python
#Question 10
def convertion_duree(duree):
    heures=duree//3600
    secondes=duree%3600
    minutes=secondes//60
    secondes=secondes%60
    return "{} heures, {} minutes et {} secondes".format(heures, minutes, secondes)

print('\nQuestion 10')
print(convertion_duree(duree_achats(panier)))
```

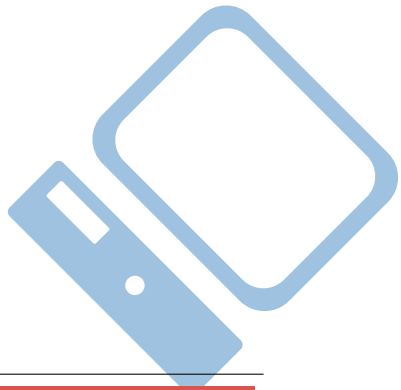
Question 11

```python
#Question 11
produits_commandes=[]
def commander(produit):
    produits_commandes.append(produit)

def defiler(panier):
    commander(panier[0])
    panier.pop(0)

def commander_panier(panier):
    while len(panier)>0:
        defiler(panier)

print('\nQuestion 11')
print('Panier avant commande')
print(panier)
```



```
17 commander_panier(panier)
18 print('Panier après commande')
19 print(panier)
20 print('Produits commandés')
21 print(produits_commandes)
```