

## 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.



```

1 def convert_bin(decimal):
2     binaire=[]
3     r=decimal
4     while r>1:
5         .....
6         .....
7         binaire.append(r)
8         r=q
9     binaire.append(r)
10    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.

```

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

```

On rappelle que sous forme polynomiale, un nombre binaire quelconque est exprimé par :  $N = \sum_0^n \alpha_i \cdot 2^i$ , 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 enfiler 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 mesurer 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 stockés 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 `conversion_duree(duree)` qui prend en entrée une durée en seconde et qui retourne le résultat sous le format suivant :

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

où heures, minutes, secondes correspondent à la conversion de durée. **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 :

```
1 produits_commandes=[]
2 def commander(produit):
3     produits_commandes.append(produit)
4
5 def defiler(panier):
6     .....
7     .....
8
9 def commander_panier(panier):
10     while len(panier)>0:
11         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

