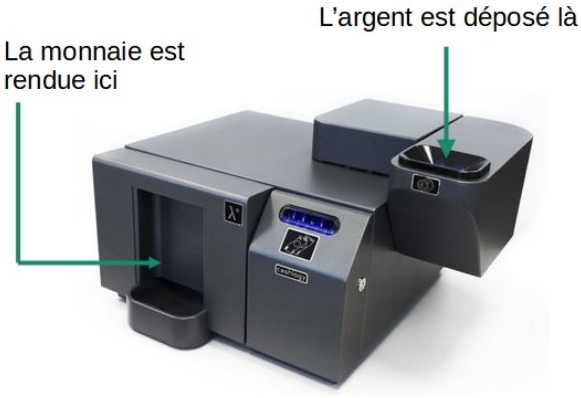
Algorithmes gloutons

I- Rendu de monnaie

Vous êtes employé par l'entreprise CashExpress, spécialisée dans la vente de caisses enregistreuses automatiques : pour éviter les contacts entre les employés et la clientèle, le client dépose directement l'argent dans un automate qui se charge de lui rendre la monnaie



On suppose que la monnaie peut-être rendue avec toutes les pièces et billets existant en euros :
[200, 100, 50, 20, 10, 5, 2, 1, 0.5, 0.2, 0.1, 0.05, 0.01]
(En effet, les billets de 500€ ne sont plus produits depuis 2018)

1) Un client vous présente un billet de 50€ pour payer sa chocolatine a 0,85€, indiquez le détail du rendu de monnaie que vous allez faire.

Un algorithme évident afin de rendre systématiquement correctement la monnaie, serait de rendre uniquement des pièces de 1 cent.
Cela présente 2 inconvénients majeurs :

- il est nécessaire de prévoir un très gros volume de pièce de 1 cent afin de pouvoir continuer à rendre la monnaie tout au long de la journée
- Vous risquez d'irriter vos clients

2) Proposez une stratégie vous permettant systématiquement de rendre la monnaie en utilisant un nombre minimal de pièces/billets.

3) Écrire la fonction **rendu_monnaie(montant)** qui retourne un *dictionnaire* contenant le détail du rendu de monnaie pour un **montant** donné.
Exemple : Le client a payé sa baguette à 1,05€ avec un billet de 5€ → l'automate doit lui rendre 3,95€
rendu_monnaie(3,95) →
{'200' :0,'100' :0,'50' :0,'20' :0,'10' :0,'5' :0,'2' :1,'1' :1,'0.5' :1,'0.2' :2,'0.1' :0,'0.05' :1,'0.01' :0}
En effet, le rendu optimal est 2€ + 1€ + 50ct + 2x20ct + 5ct

Une boulangerie basée en Absurdistan souhaite se procurer une de vos machine. Leur monnaie est le Plouk, dont le taux de change est proche de l'Euro : 1 Pk = 1 €.
Leur monnaie est différemment échelonnée : [200, 100, 30, 20, 5, 2, 1, 0.3, 0.1, 0.05, 0.01]

4) En adaptant la fonction **rendu_monnaie(montant)** aux différentes valeurs disponibles dans ce pays, quel retour propose-t-elle pour 45 Pk ?
Ce rendu est-il optimal ?

II- Les fractions égyptiennes

Si les égyptiens ne connaissaient pas la notation décimale, ils connaissaient les fractions, sous une forme particulière (voir Wikipédia) : leurs fractions étaient de la forme $\frac{1}{p}$ où $p \in \mathbb{N}^*$. On parle de fraction unitaire.

Ils pouvaient donc écrire n'importe quel nombre rationnel (un nombre qui peut s'écrire comme le quotient de 2 nombres entiers) en le décomposant d'une infinité de façons à l'aide de fractions unitaires.

Par exemple $\frac{2}{5}=\frac{1}{5}+\frac{1}{6}+\frac{1}{30}=\frac{1}{5}+\frac{1}{8}+\frac{1}{20}+\frac{1}{40}$.

Cette solution les satisfaisait probablement d'autant plus qu'il n'est pas sûr qu'ils savaient que certains nombres ne pouvaient être rationnels, $\sqrt{2}$ ou π !

Il vous faudrait écrire un algorithme glouton **egypte(nombre)** qui décompose n'importe quel nombre 0<x<1 en somme de fractions unitaires…

Remarque : cette façon de décomposer un nombre décimal devrait vous rappeler quelque chose, puisque c'est de cette façon que l'on détermine l'écriture binaire d'un décimal, en utilisant des fractions unitaires de la forme $\frac{1}{2^n}$

III- Le voleur

Marcel Lupin est un voleur prévoyant. Avant de perpétrer ses larcins, il prend soin de lister les objets de valeur présents chez ses futures malheureuses victimes. Il n'emporte sur les lieu qu'un seul sac de 50kg, il doit donc être sélectif !
Il établit donc une liste (enfin, nous dirions plutôt *dictionnaire* mais Marcel n'est pas informaticien) des objets présents, en indiquant tout d'abord leur masse en kg puis leur valeur en Brouzoufs (c'est la monnaie d'Andorrsbourg, riche petite principauté voisine de l'Absurdistan dans laquelle se déroule notre histoire).
Cette liste est disponible ici : https://bernon.fr/nsi/voleur.txt)

Proposez une fonction **a_voler(dictionnaire)** qui pour un **dictionnaire** donné retourne la liste des items à voler en priorité afin d'avoir la plus grande valeur dans le sac.
Cette fonction calcule également la valeur du contenu du sac.