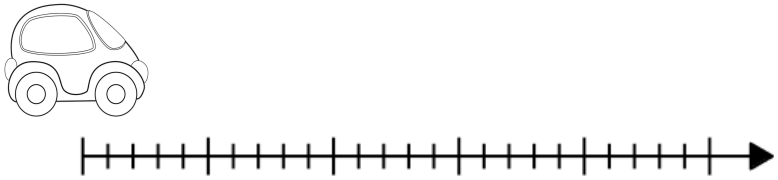
Algorithmes gloutons – EXERCICE 2

LES STATIONS D'ESSENCE: fichier **stations.py**

Un automobiliste part en vacances et doit parcourir un long trajet. Il prend la route avec le plein de carburant. Son véhicule peut parcourir une distance maximale d avec son plein. La route empruntée comporte n stations services: S_0, S_1, S_2, ...,S_{n-1} rangées dans l´ordre rencontré pendant le parcours. La première est à une distance d_0 du départ, la deuxième est à une distance d_1 de la première etc. Le point d´arrivée est à une distance d_n de la dernière station.



L´objectif de l´automobiliste est de s´arrêter le moins souvent possible...
Le fichier `station.py` contient la fonction essence qui va déterminer de manière gloutonne les stations dans lesquelles il devra s´arrêter.
Elle prend deux paramètres: une liste de distances entre les stations et la distance maximale qui peut être parcourue avec un plein.

```
1. def essence(liste, dmax):
2.     n=len(liste)
3.     d=dmax
4.     stations=[]
5.     i=0
6.     while i!=n:
7.         while i<n and liste[i]<=d:
8.             d=d-liste[i]
9.             i=i+1
10.             stations.append(i-1)
11.             d=dmax
12.         return stations
```

Test de la fonction gloutonne:

1) Commençons par créer une fonction **`somme(liste)`** qui permet de calculer la somme de tous les éléments de la liste passée en paramètre.

2) Créer ensuite la variable **`trajet`** qui contiendra la distance à parcourir et la variable **`réservoir`** qui contiendra le nombre de kilomètres que la voiture peut parcourir avec un plein. Ces deux variables sont des entiers entrés par l´utilisateur.

Création de la liste des distances entre stations. Cette liste contiendra des distances aléatoires entre les stations comprises entre 25 et 50 kilomètres.
Exemple de liste créée pour une distance à parcourir de 250 km: **`[57, 44, 25, 27, 58, 39]`**

3) Après importation du module **`random`**, commençons par créer une liste **`tab`** avec un nombre entier (`randint(25,50)`) aléatoire compris entre 25 et 50.

4) A l´aide d´une boucle `while`, ajouter des distances aléatoires entre 25 et 50 *tant que* la somme des distances est inférieure au trajet à parcourir. C´est ici que la fonction **`somme(liste)`** est utile.

5) Il faut maintenant modifier la dernière valeur de la liste de manière à ce que la somme totale des éléments de la liste soit égale au trajet à parcourir. On accède au dernier élément de la liste par: **`tab[-1]`**

Testez votre fonction. Si tout fonctionne, vous pouvez maintenant prendre la voiture et partir enfin sur la route des vacances.

Stéphane Glouton
Frédéric Glouton