

DEVOIR EN TEMPS LIBRE D'INFORMATIQUE N° 1

Exercice : Quelques boucles autour des nombres

1. Algorithme de Kaprekar Etant donné un entier n de 3 chiffres, non tous égaux, on considère le processus suivant :

- On calcule P , le plus grand nombre constitué avec les mêmes chiffres que n .
- On calcule q , le plus petit nombre constitué avec les mêmes chiffres que n .
- On calcule la différence $P - q$
- On réitère le processus jusqu'à retrouver un nombre déjà obtenu. A noter que si au cours du processus, on tombe sur un nombre à 2 chiffres, on complète l'écriture de ce nombre avec autant de 0 qu'il faut à gauche

Par exemple, si on part de $n = 242$. Après un itération, on arrive à : $422 - 224 = 198$. Puis à $981 - 189 = 792$, puis $972 - 279 = 693$, puis $963 - 369 = 594$ puis $954 - 459 = 495$ puis $954 - 459 = 495$ et on arrête l'algorithme.

(a) Si n est un nombre la séquence de commandes suivante, crée la liste $Liste_n$ de ses chiffres :

```
1 mot_n = str(n)
2 Liste_n = [int(x) for x in mot_n]
```

Une liste de nombres L étant donnée, les listes $L1$ et $L2$ créées par les commandes suivantes contiennent les mêmes nombres que L (avec le même nombre d'occurrences) dans l'ordre croissant pour $L1$, dans l'ordre décroissant pour $L2$

```
1 L.sort()
2 L1 = L[:]
3 L2 = L[::-1]
```

Ecrire une séquence d'instructions qui :

- Demande à l'utilisateur d'entrer un nombre n à trois chiffres non tous égaux
- Calcule le nombre obtenu après une itération du processus décrit
- Répète ce processus jusqu'à avoir trouver un nombre déjà obtenu. Pour ce dernier point, on pourra créer une liste *ListeDesNombres* initialisée à $[n]$ et qui sera complétée à chaque itération par le nombre x obtenu à l'aide de la commande : *ListeDesNombres.append(x)*. Enfin pour savoir si un nombre y a déjà été obtenu, il suffit de faire le test :

```
1 if y in ListeDesNombres :
```

(b) Faites le test avec différentes valeurs de n : au moins avec 254, 484, 667, 148. Que peut-on conjecturer ?

(c) Effectuez le même travail pour un nombre à 4 chiffres. Pour un nombre à 2 chiffres.

2. Produit des chiffres d'un nombre

- (a) Etant donné un entier n , écrire une séquence d'instructions permettant de calculer le produit de ses chiffres
- (b) Réciproquement, quels sont les nombres qui peuvent s'écrire comme produit des chiffres d'un entier ?
- (c) Pour un entier p donné, écrire une séquence d'instructions qui donne la variable *resultat* affectée de la valeur :
- le booléen *False* si p ne peut pas s'écrire comme produit des chiffres d'un entier
 - le plus petit entier n dont le produit des chiffres vaut n dans le cas contraire.