

L'objectif de ce TD est de continuer à s'entraîner à écrire des fonctions récursives. Cette fois-ci, nous allons utiliser notre ordinateur pour explorer de façon systématique toutes les possibilités offertes par des lancers de dés à six faces. Dans ce TD, toutes les fonctions demandées **doivent être récursives**.

Exercice 1 : énumération des lancers possibles

Question 1

En guise d'échauffement, implémenter une fonction `lance(nb_des)` renvoyant la somme des valeurs du nombre de dés demandé lancés aléatoirement (une seule fois).

Question 2

Sans lien avec la question précédente, on souhaite maintenant énumérer tous les lancers possibles pour un nombre de dés donné. Il n'est donc plus question d'aléatoire ici. Par énumération nous entendons ici afficher sur la sortie standard tous les lancers de dés possibles.

Par exemple pour 2 dés on cherche à afficher les lancers suivants. La première colonne représente le premier dé, et la seconde le deuxième.

```
1 1 1
2 1 2
3 1 3
4 ...
5 2 1
6 2 2
7 ...
8 6 5
9 6 6
```

Pour ce faire, on utilise un tableau dynamique (donc une `list` Python) `des` dont chaque case correspond à un dé. Initialement, le tableau dynamique est rempli de 0 pour indiquer qu'aucun dé n'a encore été tiré. Il sera ensuite rempli avec des valeurs entre 1 et 6 au cours de l'énumération.

On effectue une récurrence sur le nombre de dés non encore lancés.

Implémenter une fonction récursive `enumere_lancers_rec(des, des_restants)`, où `des` contient un lancer partiel et `des_restants` le nombre de dés restant à lancer, qui affiche sur la sortie standard tous les lancers possibles.

Implémenter une fonction `enumere_lancers(nb_des)` qui appelle `enumere_lancers_rec(des, des_restants)` avec les paramètres initiaux permettant d'énumérer tous les lancers possibles pour le nombre de dés spécifié.

Question 3

Implémenter une fonction `enumere_lancers_somme(nb_des, somme)` affichant tous les lancers comme `enumere_lancers(nb_des)` mais qui "encadre" l'affichage des lancers dont la somme vaut `somme` par "****".

Afin de factoriser notre code, on utilisera la fonction `enumere_lancers_rec` de la question précédente. Nous ajouterons le nécessaire à cette fonction pour pouvoir répondre à cette question tout en préservant le comportement de la question précédente.

Exercice 2 : compter le nombre de lancers

Question 1

Implémenter une fonction `compte_occurrence_somme(nb_des, somme)` renvoyant le nombre de lancers du nombre de dés donné atteignant la somme demandée. On essaiera d'éviter d'énumérer toutes les combinaisons possibles.

Question 2

On suppose disposer d'une fonction `fonction_validation` prenant un tableau dynamique de dés en argument. Cette fonction classe les lancers de dés en valides ou invalides selon un certain critère arbitraire : elle renvoie `True` lorsqu'un lancer doit être considéré comme valide et `False` dans le cas contraire.

Implémenter une fonction `compte_lancers_valides(nb_des, fonction_validation)` renvoyant pour le nombre de dés donné le nombre de lancers valides selon la fonction de validation donnée.

Exercice 3 : est-ce que 1, 2, 4 est différent de 4, 1, 2? (pour aller plus loin)

Quand on joue aux dés, et qu'on les lance tous d'un coup, "l'ordre" n'a aucune importance. Autrement dit, avec 3 dés par exemple, le lancer 1, 2, 4 est équivalent au lancer 4, 1, 2.

Question 1

Reprendre la question 2 de l'exercice 1 pour énumérer seulement les lancers distincts (relativement à la permutation des dés).

Question 2

Combien de lancers distincts existent avec `nbdes` dés?