

# Combinatoire

Dans ce TP, on veut programmer certains objets combinatoires vus en cours.

Attention, les objets combinatoires ayant des tailles qui deviennent vite très grandes, on fera attention aux exemples choisis (une quinzaine, voire une vingtaine d'éléments).

**Exercice 1.** On commence par la liste des permutations de éléments d'une liste.

Pour cela, on va définir des rotations successives; par exemple, une rotation de  $[1, 2, 3, 4]$  est  $[2, 3, 4, 1]$ . On admet qu'en calculant toutes les rotations possibles de tous les éléments de la liste, puis les rotations de ces rotations à partir du deuxième élément, *etc.*, on obtient exactement les permutations.

Par exemple, on peut calculer la liste des permutations de  $[1, 2, 3]$  en faisant :

- On commence par les rotations à partir de  $k = 0$  :  $[1, 2, 3]$ ,  $[2, 3, 1]$  et  $[3, 1, 2]$
- Pour chacune d'entre elles, on fait les rotations à partir de  $k = 1$  :  $[1, 2, 3]$ ,  $[1, 3, 2]$ ,  $[2, 3, 1]$ ,  $[2, 1, 3]$ ,  $[3, 1, 2]$ ,  $[3, 2, 1]$

**Exercice 2.** On veut maintenant calculer l'ensemble des parties.

Écrire une fonction qui prend une liste en paramètre, et renvoie la liste de ses parties. On utilisera la propriété suivante : si  $x \in E$ , les parties de  $E$  sont exactement celles de  $E \setminus \{x\}$ , plus celles de  $E \setminus \{x\}$  auxquelles on ajoute  $x$ .

**Exercice 3.** Dédire de la fonction précédente une fonction calculant la liste des combinaisons de  $p$  objets parmi une liste  $l$ .

**Exercice 4.** Dédire des fonctions précédentes une fonction calculant la liste des  $p$ -arrangements d'une liste  $l$ .