

## Programmation par contraintes en Oz

### Exercice 1

On cherche à écrire un programme pour optimiser le transport d'essence par des camions-citernes. Le nombre de camions disponibles avec des capacités différentes est décrit par un enregistrement qui associe des entiers (le nombre de camions) à des capacités. Par exemple,

```
cap(40:6 35:8 25:10 10:5)
```

indique un parc de camions consistant en 6 camions avec une capacité de  $40m^3$  chacun, 8 camions à  $35m^3$ , 10 camions à  $25m^3$ , et 5 camions à  $10m^3$ .

Un camion avec une capacité supérieure à  $30m^3$  nécessite deux chauffeurs, tandis que les camions plus petits peuvent opérer avec un seul chauffeur.

Écrire une fonction Oz, en utilisant des contraintes à domaines finis, qui prend en argument une description d'un parc de camions comme au-dessus, et un volume d'essence à transporter exprimé en nombre de  $m^3$ . Le résultat de la fonction doit être soit nil dans le cas où il n'y a pas de solution, soit une liste qui a comme seul élément l'enregistrement qui associe à chaque capacité le nombre de camions de cette capacité qui permet de transporter le volume de liquide avec un minimum de chauffeurs.

Par exemple, pour le parc comme donné au-dessus, et un volume de  $381m^3$  d'essence à transporter, une solution optimale qui nécessite 17 chauffeurs est

```
cap(40:4 35:0 25:9 10:0)
```

### Exercice 2

Une *séquence magique* de longueur  $n$  est une séquence d'entiers  $x_1, \dots, x_n$  entre 0 et  $n - 1$  telle que pour tout  $i = 0, \dots, n - 1$  le nombre  $i$  paraît exactement  $x_{i+1}$  fois dans la séquence.

Un exemple d'une séquence magique de longueur 7 est

3, 2, 1, 1, 0, 0, 0

car 0 paraît  $x_1 = 3$  fois dans la séquence, 1 paraît  $x_2 = 2$  fois dans la séquence, ..., 6 paraît  $x_7 = 0$  fois dans la séquence.

Écrire une fonction en Oz, en utilisant les contraintes à domaines finis, qui prend un entier  $n$  en argument et qui envoie la liste de toutes les séquences magiques de longueur  $n$ .

Voici les résultats attendu de votre fonction pour quelques valeurs de  $n$  :

$n$	résultat
1	nil
2	nil
3	nil
4	[[1 2 1 0] [2 0 2 0]]
5	[[2 1 2 0 0]]
6	nil
7	[[3 2 1 1 0 0 0]]

*Conseil* : Utiliser le propagateur FD.exactly.