Programmation par contraintes en Oz

Exercice 1
Écrire en Oz un programme qui permet de résoudre le puzzle suivant : Quatre maisons, numérotées de 1 (à gauche) à 4 (à droite), se trouvent le long d'une rue. Chacune des maisons est habitée par une personne de nationalité différente (parmi Danois, Italien, Britannique, et Allemand). Toutes les maisons sont peintes d'une couleur différente (parmi rouge, blanc, noir, et bleu). Chacun des habitants a une boisson préférée différente (parmi thé, lait, bière, et vin) et un plat préféré différent (parmi pizza, des gaufres, chocolat, et des pâtes), et pratique un sport différent (parmi basket-ball, tennis, squash, et badminton). Les indications sont :

1. La première maison est noire, et la deuxième n'est pas bleue.
2. Le Britannique joue au basket-ball.
3. L'Italien habite à droite (à une distance quelconque) de la personne qui mange du chocolat.
4. L'Allemand habite à côté du mangeur de gaufres.
5. Il y a deux maisons entre le joueur de basket-ball et le buveur de vin.
6. Il y une maison entre le joueur de badminton et le buveur de bière.
7. Le mangeur de chocolat ne boit pas de la bière, et le joueur de tennis ne boit pas du vin.
8. Le Britannique et l'Italien ne sont pas des voisins.
9. La distance entre la maison bleue et la maison rouge est la même que la distance entre les maisons du joueur de tennis et du joueur de squash.
10. Le joueur de tennis, le mangeur de pizza, et l'allemand habitent tous dans des maisons différentes, dont aucune n'est rouge.
11. Le Danois boit du thé.

Votre programme doit trouver pour chacune des couleurs de maisons, et pour chaque boisson, chaque plat, chaque nationalité et chaque sport le numéro de la maison associée.

Le puzzle a une solution unique.

Indication : Créer une variable à domaine fini pour chaque couleur, chaque boisson, chaque plat, chaque nationalité et chaque sport.

Exercice 2

Un *triangle de distances* de taille n est un triangle de largeur n et profondeur n tel que ses positions sont occupées par des valeurs différentes entre 1 et $\frac{n*(n+1)}{2}$, et tel que toute valeur (sauf sur la première ligne) est la distance absolue entre les deux valeurs qui se trouvent sur la ligne au-dessus, directement à la gauche et directement à la droite. Par exemple, un triangle de distance de taille 5 possible est le suivant :

$$\begin{array}{ccccccccc} 6 & & 14 & & 15 & & 3 & & 13 \\ & 8 & & 1 & & 12 & & 10 & \\ & & 7 & & 11 & & 2 & & \\ & & & 4 & & 9 & & & \\ & & & & 5 & & & & \end{array}$$

Puisque $8 = |6 - 14|$, $1 = |14 - 15|$, $12 = |15 - 3|$, $\ldots$, $5 = |4 - 9|$.

1. Écrire un programme en Oz, en utilisant les contraintes de domaine fini, qui génère tous les triangles de distance de taille 5.

2. Écrire une procédure en Oz, en utilisant les contraintes de domaine fini, qui prend en entrée un entier n et qui génère tous les triangles de distance de taille n. Tester au moins avec les valeurs pour n entre 1 et 7.

Indication : Créer d'abord un triangle de variables à domaine fini comme une liste qui contient des listes de variables à domaine fini. Servez vous des fonctions de la bibliothèque standard :

- {List.make N} envoie une liste avec N variables fraiches
- {FD.list N spec}, où spec est la spécification d'un domaine, envoie une liste avec N variables qui ont toutes le domaine spec.

N'oubliez pas que FD.distribute attend un vecteur avec des variables au niveau 1, c-à-d il va falloir aplatir la structure créée avant de la passer à FD.distribute.