Option Informatique en Sup MPSI
TP : Sudoku, un corrigé

Question 1 • Récréation : complétez la grille !

Question 2 • Vous avez le choix entre le très concis `it_list (prefix @) []` et le très classique :

```
let rec flat = function
 | [] -> []
 | t::q -> t @ (flat q) ;;
```

Question 3 • La fonction `intervalle` permet une grande concision :

```
let list_of_bande h g d m =
 map (fun j -> m.(h).(j)) (intervalle g d) ;;
```

Question 4 • Pour chaque valeur i de l'indice de ligne, nous construisons la liste des cases de la bande définie par i, g et d ; ensuite, il reste à remettre à plat la liste de listes obtenue, en utilisant `falat` :

```
let list_of_zone h b g d m =
 flat (map (fun i -> list_of_bande i g d m) (intervalle h b)) ;;
```

Question 5 • Il suffit d'appliquer la fonction `list_of_zone` aux bons argument. Le bloc numéro k contient les cases (i, j) avec $g \leqslant i \leqslant g + 2$ et $h \leqslant j \leqslant h + 2$, où $g = 3\lfloor k/3 \rfloor$ et $h = 3(k \bmod 3)$.

```
let list_of_bloc k m = let g = 3*(k/3) and h = 3*(k mod 3) in
 list_of_zone (g) (g+2) (h) (h+2) m ;;
```

Question 6 • Le plus simple est de trier la liste ℓ et de la comparer à la liste ≪canonique≫ :

```
let bonne_liste l = sort__sort (prefix <) l = intervalle 1 9 ;;
```

Question 7 • Il suffit de vérifier que chacune des trois listes associées respectivement à la ligne, à la colonne, et au bloc de la case visée, est correcte.

```
let bon_sudoku m = let l08 = intervalle 0 8 in
 let lignes = map (fun i -> list_of_bande i 0 8 m) l08
 and colonnes = map (fun j -> list_of_zone 0 8 j j m) l08
 and blocs = map (fun k -> list_of_bloc k m) l08 in
 for_all bonne_liste lignes && for_all bonne_liste colonnes
 && for_all bonne_liste blocs ;;
```

Personellement, j'aime beaucoup la variante suivante, qui n'est pas plus courte, mais est certainement plus mystérieuse...

```
let f = it_list (prefix &) true ;;

let bon_sudoku m = let l08 = intervalle 0 8 in
 let lignes = map (fun i -> list_of_bande i 0 8 m) l08
 and colonnes = map (fun j -> list_of_zone 0 8 j j m) l08
 and blocs = map (fun k -> list_of_bloc k m) l08 in
 f(map f (map (map bonne_liste) [lignes;colonnes;blocs])) ;;
```

Celle-ci est plus courte et plus claire, ce qui prouve que l'on peut avoir le beurre *et* l'argent du beurre :

```
let bon_sudoku m =
 let f1 = fun i -> list_of_bande i 0 8 m
 and f2 = fun j -> list_of_zone 0 8 j j m
 and f3 = fun k -> list_of_bloc k m in
 for_all bonne_liste (flat(map (fun f -> map f l08) [f1;f2;f3])) ;;
```

▶ Pour les dernières questions, consultez le site web !