Option Informatique en Sup MPSI
Calculs dans l'algèbre des parties finies ou cofinies de $\mathbb{N}$

▶ Les premières questions de ce problème vous demandent de rédiger quelques fonctions simples, dont certaines font d'ailleurs partie de la bibliothèque Caml.

▶ Pour les évaluations de coûts, l'unité est le *Cons*, c'est-à-dire l'opérateur Caml `::` (qui permet aussi bien d'isoler, dans une liste, la tête et la queue, que de construire une nouvelle liste à partir d'une tête et d'une queue). La longueur d'une liste ℓ est notée $|\ell|$.

▶ Un *prédicat* est une fonction à valeurs booléennes. On dit que le prédicat p est *satisfait* par l'objet x lorsque $p(x) =$ `true`

Question 1 • Rédigez une fonction :

```
mem : 'a -> 'a list -> bool
```

spécifiée comme suit : `mex x l` indique si x est présent dans la liste ℓ ; le coût en sera un $\mathcal{O}(|\ell|)$.

Question 2 • Rédigez une fonction :

```
filtre : ('a -> bool) -> 'a list -> 'a list
```

spécifiée comme suit : `filtre p l` construit la liste des éléments de ℓ qui satisfont le prédicat p ; le coût en sera un $\mathcal{O}(|\ell|)$.

Question 3 • Rédigez une fonction :

```
intersection_simple : 'a list -> 'a list -> 'a list
```

spécifiée comme suit : `intersection_simple l1 l2` construit la liste des éléments communs aux deux listes ℓ_1 et ℓ_2 ; le coût en sera un $\mathcal{O}(|\ell_1| \times |\ell_2|)$. Vous ferez en sorte que, si aucune de ces listes ne contient de doublon, le résultat n'en contienne pas non plus.

Question 4 • Rédigez une fonction :

```
union_simple : 'a list -> 'a list -> 'a list
```

spécifiée comme suit : `union_simple l1 l2` construit la liste des éléments qui sont présents dans l'une au moins des deux listes ℓ_1 et ℓ_2 ; le coût en sera un $\mathcal{O}(|\ell_1| \times |\ell_2|)$. Vous ferez en sorte que, si aucune de ces listes ne contient de doublon, le résultat n'en contienne pas non plus.

Question 5 • Rédigez une fonction :

```
for_all : ('a -> bool) 'a list -> bool
```

spécifiée comme suit : `for_all p l` rend la valeur `true` si et seulement si tous les éléments de la liste ℓ satisfont le prédicat p ; le coût en sera un $\mathcal{O}(|\ell|)$.

▶ Une partie A de $\mathbb{N}$ est *cofinie* si son complémentaire est une partie finie de $\mathbb{N}$. On note $\mathcal{W}$ l'ensemble des parties finies ou cofinies de $\mathbb{N}$.

Question 6 • Montrez que $\mathcal{W}$ est la plus petite famille de parties de $\mathbb{N}$ qui contienne les parties finies et qui soit stable pour les opérations booléennes : union, intersection, complémentation. Bien entendu, ≪petite≫ est à comprendre au sens de l'inclusion.

▶ Pour décrire les éléments de $\mathcal{W}$, on définit le type Caml suivant :

```
type ficof = Finie of int list | Cofinie of int list;;
```

L'interprétation de ce type est évidente : `Finie [2;8;1]` désigne la partie $\{1, 2, 8\}$ de $\mathbb{N}$ tandis que `Cofinie [8;1;2]` désigne le complémentaire de cette partie. Comme on le constate sur ces exemples, l'ordre d'énumération dans les listes est indifférent. En revanche, il ne doit pas y avoir de doublon : par exemple, la partie $\{3, 6, 8\}$ ne devra pas être représentée par `Finie [6;3;6;8]`.

Question 7 • Rédigez une fonction :

```
appartient : int -> ficof -> bool
```

spécifiée comme suit : `appartient n p` indique si le naturel n appartient à la partie finie ou cofinie p de $\mathbb{N}$.

Question 8 • Rédigez une fonction :

```
contient : ficof -> ficof -> bool
```

spécifiée comme suit : `contient p q` indique si la partie finie ou cofinie p de $\mathbb{N}$ contient la partie finie ou cofinie q de $\mathbb{N}$.

Question 9 • Rédigez une fonction :

```
complement : ficof -> ficof
```

spécifiée comme suit : `complement p` calcule le complémentaire de la partie finie ou cofinie p de $\mathbb{N}$.

Question 10 • Rédigez une fonction :

```
union : ficof -> ficof -> ficof
```

spécifiée comme suit : `union p q` calcule la réunion des parties finies ou cofinies p et q de $\mathbb{N}$.

Question 11 • Rédigez une fonction :

```
intersection : ficof -> ficof -> ficof
```

spécifiée comme suit : `intersection p q` calcule l'intersection des parties finies ou cofinies p et q de $\mathbb{N}$.

Question 12 • Donnez une estimation du coût de vos fonctions, en nombre de *Cons*, et en fonction des longueurs des listes représentant les parties finies ou cofinies manipulées.

Question 13 • Quelles propositions faites-vous pour abaisser ces coûts ?

Question 14 • Les fonctions que vous avez rédigées ne peuvent permettre que les calculs dans l'algèbre des parties finies ou cofinies de $\mathbb{N}$. Quelle propriété du langage Caml permet à peu de frais de travailler dans l'algèbre des parties finies ou cofinies d'un autre ensemble ? Qu'en est-il alors des possibilités de diminution des coûts évoquées à la question précédente ?

FIN