




On se propose ici de créer une structure de donnée permettant de représenter un ensemble infini sous la forme d'une fonction de `'a -> bool` déterministe et qui **termine toujours**. On fixe dans cet exercice X un ensemble quelconque. On définit donc le type suivant en OCaml :


```
type 'a set = 'a -> bool;;
```


 **Question 0** Expliciter la bijection entre $\mathcal{P}(X)$ et $\{0, 1\}^X$


 **Question 1** Donner une fonction `make_finite: 'a list -> 'a set` qui à une liste finie renvoie l'ensemble de ses valeurs


 **Question 2** Construire des objets de type `int set` pour représenter les ensembles suivants :

- L'ensemble des nombres pairs
- \mathbb{P} , l'ensemble des nombres premiers
- L'image d'une fonction fixée f positive et strictement croissante

 **Question 3** Donner le code d'une fonction `union` qui réalise l'union de deux ensembles.

 **Question 4** Donner un ensemble que l'on ne pourra pas représenter par notre structure. Le nombre d'ensembles non représentable est-il fini ? Dénombrable ? Indénombrable ?

 **Question 5** Soit `val P: int set set`, montrer qu'il existe un $N \in \mathbb{N}$ tel que pour tout `val x: int set`, `P x` ne regarde que les entrées inférieure à N dans `x`. Est-ce vrai avec `val P: (int -> int) set` ?

 **Question 6** Écrire une fonction `val f: (int set set) -> int set` qui à un `int set set` non vide associe un de ces éléments.