

Interrogation écrite 3

INF 201 — IMA3&4 — 27/03/2026 — 20 minutes

Exercice 1. (*Inspiré du projet*) (/10) On décide d'implémenter un type ensemble de la façon suivante:

```
type 'a ensemble = 'a list;
```

Question 1. (/2) Implémenter une fonction `appartient` qui teste l'appartenance d'un élément à un ensemble.

```
val appartient : 'a -> 'a ensemble -> bool = <fun>
```

Question 2. (/3) Montrer que la fonction `appartient` termine.

Question 3. (/2) Implémenter une fonction `ajoute` qui ajoute un élément à un ensemble. On rappelle que dans un ensemble les éléments ne se répètent pas !

```
val ajoute : 'a -> 'a ensemble -> 'a ensemble = <fun>
```

Question 4. (/3) Implémenter une fonction `intersection` qui calcule l'intersection de deux ensembles. On rappelle que $A \cap B = \{x : x \in A \text{ et } x \in B\}$.

```
val intersection : 'a ensemble -> 'a ensemble -> 'a ensemble = <fun>
```

Exercice 2. (*Inspiré du partiel*) (/10) On rappelle la définition des types `grandeurphy` et `unite`:

```
type grandeurphy = Long | Masse | Temps
```

```
type unite = U of grandeurphy | Rapp of unite*unite | Prod of unite*unite | Puiss of unite*int | Aucune
```

On définit les constantes suivante:

```
let accel = Rapp(U Long, Puiss(U Temps, 2))
```

```
let force = Prod(accel, U Masse)
```

Question 5. (/2) En déduire une expression de l'unité de la pression, on rappelle que $P = F/S$, avec P la pression, F la force et S la surface.

```
let surface =
```

```
let pression =
```

On suppose qu'on dispose d'une fonction `homogene` qui est en mesure de dire si deux expressions du type `unite` sont de la même dimension ou non.

On veut écrire une fonction `somme` qui prend une liste de nombres flottants accompagnés de leurs `unite` et effectue la somme des termes si ceux si sont de la même unité ! Dans ce cas on renverra un couple avec la somme et son unité, sinon, si tous les termes ne sont pas de la même unité on renverra 0. et `Aucune` pour la dimension.

Question 6. (/2) Donner le profil de `somme`

Question 7. (/2) Compléter les exemples:

```
let v = Rapp(U Long, U Temps);;
```

```
let a = Rapp(v, U Temps);;
```

```
let l1 = [(10., v);(3., v);(1., v)];;
```

```
let l2 = [(100., v);(9.81, a)];;
```

```
somme l1 =
```

```
somme l2 =
```

Question 8. (/4) Implémenter `somme`