Interrogation écrite 1

INF 201 — IMA3&4 — 30/01/2026 — 15 minutes

Exercice 1. (/4) À chaque fois, donner le profil de la fonction:

```
let f x y z = x && y || z;;
```

```
bool -> bool -> bool -> bool
```

```
let f a b = if a then b else 0;;
```

```
bool -> int -> int
```

```
let f x y z = if x > 0. then y + 1=2 else z;;
```

```
float -> int -> bool -> bool
```

```
let f p x y = if p x then y x else print_int x;;
```

```
(int -> bool) -> int -> (int -> unit) -> unit
```

Exercice 2. (/2) Cours:

```
type saison = Printemps | Ete | Automne | Hiver;;
```

Comment se nomme le type `saison` ?

enum

Comment appelle t-on les valeurs `Printemps`, `Ete`, `Automne`, `Hiver` ?

Constructeurs

Exercice 3. (/4)

a	b	$a \wedge b$	$a \vee b$	$\neg a \wedge \neg b$	$a \wedge b \vee (\neg a) \vee (\neg b)$
V	V	V	V	F	V
V	F	F	V	F	V
F	V	F	V	F	V
F	F	F	F	V	V

Exercice 4. (/3)

On définit la fonction signe de la manière suivante:

$$\begin{array}{rcl} \text{sgn} & : & \mathbb{R} \to \mathbb{Z} \\ \text{sgn}(x) & = & \begin{cases} -1 & \text{si } x < 0 \\ 0 & \text{si } x = 0 \\ 1 & \text{si } x > 0 \end{cases} \end{array}$$

Proposer une implémentation en OCaml de cette fonction en utilisant `if...then...else` **OU** le filtrage par motif:

```
let sgn x = if x < 0. then -1
            else if x = 0. then 0
            else 1;;
```

Exercice 5. (/8) On souhaite calculer l'heure qu'il sera après avoir ajouté une durée à une heure donnée. On travaille sur une **horloge 24 heures**. Par exemple si on part de 22h50, et on ajoute 135 minutes alors on obtient 1h05 (le lendemain).

Proposer un type `heure` pour représenter les heures: (ne pas oublier d'indiquer la restriction)

```
type heure = int;; (* Entre 1 et 23 *)
```

Proposer un type `minute` pour représenter les minutes: (ne pas oublier d'indiquer la restriction)

```
type minute = int;; (* Entre 1 et 23 *)
```

En **déduire** un type `horloge` pour représenter l'horloge:

```
type horloge = (heure*minute);;
```

Voici maintenant un exemple d'utilisation du programme qu'on souhaite écrire:

```
# en_minutes (20, 34);;
- : minute = 1234
# en_horloge 5000;;
- : horloge = (11, 20)
# ajout_duree (22,50) 135;;
- : horloge = (1, 5)
```

Donner une implémentation de `en_minutes`:

```
let en_minutes ((h,m):horloge) : minute =
  h*60 + m;;
```

Donner une implémentation de `en_horloge`:

```
let en_horloge (m:minute) : horloge =
  ((m / 60) mod 24, m mod 60);;
```

En déduire une implementation de `ajout_duree`:

```
let ajout_duree (hor:horloge) (m:minute) : horloge =
  en_horloge (((en_minutes hor) + m));;
```