Séminaire CAML
QCM n° 4
mardi 15 sept. 2015

1. **Dans la définition suivante :**

```
let f x = match x with
      pattern1 -> expression1
    | pattern2 -> expression2 ;;
```

(a) *pattern1* et *pattern2* doivent être du même type que `x`.
(b) *pattern1* et *pattern2* doivent être du même type mais pas forcément celui de `x`.
(c) *expression1* et *expression2* doivent être du même type.
(d) *pattern1* et *expression1* doivent être du même type.

---

2. **Quel est le résultat de l'évaluation de la définition suivante ?**

```
let f x = match x with
      | 0 -> 0
      | y -> 1/y ;;
```

(a) `... Unbound value y`
(b) `val f : int -> int = <fun>`
(c) `val f : 'a -> int = <fun>`
(d) Un autre message d'erreur.

---

3. **Après l'évaluation de la phrase suivante, que contient le résultat ?**

```
let switchonoff x = match x with
      "on" -> true
    | "off" -> false;;
```

(a) `val switchonoff : string -> bool = <fun>`
(b) `Warning P : this pattern-matching is not exhaustive.`
(c) `Warning U : this match case is unused.`
(d) Un message d'erreur.

---

4. **Quel est le résultat de l'évaluation de la phrase suivante ?**

```
function    2 | 4 | 6 | 8 -> 1
          | 1 | 3 | 5 | 7 | 9 -> 0
          | _ -> invalid_arg "not a digit" ;;
```

(a) `- : int -> string = <fun>`
(b) `- : int -> int = <fun>`
(c) `- : string = "not a digit"`
(d) `Unbound value function`
(e) Un autre message d'erreur.

---

5. **Si le résultat de l'évaluation d'une fonction contient le message suivant :**

```
Warning P : this pattern-matching is not exhaustive.
```

(a) Il y a au moins un cas inutile.
(b) Il manque au moins un cas.
(c) La fonction est quand même définie.
(d) La fonction n'est pas définie.

6. **Après l'évaluation de la phrase suivante, que contient le résultat ?**

```
let f = function
      0 -> 12
    | _ -> x+x
    | 1 -> 24 ;;
```

(a) *val f : int -> int = <fun>*
(b) *Warning U : this match case is unused.*
(c) *Warning P : this pattern-matching is not exhaustive.*
(d) *Error : Unbound value x*

---

7. **Que contient le résultat de l'évaluation de la phrase suivante ?**

```
let f x = function
      0 -> failwith "divisor is zero"
    | y -> if y > x then y / x
           else if x = 0 then failwith "divisor is zero"
                else x / y
    | _ -> failwith "impossible operation" ;;
```

(a) *val f : int -> int -> int = <fun>*
(b) *val f : int -> int -> string = <fun>*
(c) *Warning U : this match case is unused.*
(d) Une erreur.

---

8. **Quel est le résultat de l'évaluation de la phrase suivante ?**

```
let test = function
     0            -> "zero"
   | x when x > 0 -> "positive"
   | _            -> "negative" ;;
```

(a) *val test : 'a -> string = <fun>*
(b) *val test : int -> string = <fun>*
(c) *val function : int -> string = <fun>*
(d) Une erreur.

---

9. **Soit la fonction g définie ci-dessous. Quels sont les énoncés vrais ?**

```
let g x y = match x with
              0 -> 0
            | y -> 1
            | x -> -1 ;;
```

(a) Les deux paramètres (x et y) doivent être du même type.
(b) y peut être de n'importe quel type.
(c) x peut être de n'importe quel type.
(d) La fonction ne retourne jamais -1.
(e) Si $x \neq 0$ et $x \neq y$, la fonction renvoie -1.

---

10. **Que contient le résultat de l'évaluation de la fonction g de la question précédente ?**

(a) *val g : int -> int -> int = <fun>*
(b) *val g : int -> 'a -> int = <fun>*
(c) *... Warning P : this pattern-matching is not exhaustive.*
(d) *... Warning U : this match case is unused.*