

Séminaire CAML
QCM n° 5
jeudi 17 sept. 2015

1. Quel est le résultat de l'évaluation de l'expression suivante ?

```
((4,5.),6) ;;
```

- (a) - : int * float * int = (4, 5., 6)
- ☒ (b) - : (int * float) * int = ((4, 5.), 6)
- (c) - : float * int = ((4,5), 6)
- (d) - : (int * float) -> int = ((4, 5.), 6)
- (e) Une erreur.

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

```
let a = let b = (0, "be") in (b, "one") ;;
```

- (a) val a : int * string * string = (0, "be", "one")
- (b) val b : int * string = (0, "be")
- (c) val a : (int * string) * string = (b, "one")
- ☒ (d) val a : (int * string) * string = ((0, "be"), "one")
- (e) Une erreur.

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

```
let x = let (a,b,c) = (1,2,3)  
        in (c, string_of_int (a*b)) ;;
```

- ☒ (a) val x : int * string = (3, "2")
- (b) val x : int * int * int = (3, "1*2")
- (c) val x : int * (int * int) = (3, (1, 2))
- (d) val x : int -> string = <fun>
- (e) Une erreur.

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

```
let y = let x = (("one", (1,1.), '1'), "wow") in (x, "hum") ;;  
let (x,y) = y in y ;;
```

- (a) ((string * (int * float) * char) * string) * string = (((("one", (1, 1.), '1'), "wow"), "hum"))
- (b) (string * (int * float) * char) * string = (("one", (1, 1.), '1'), "wow")
- ☒ (c) - : string = "hum"
- (d) - : char * string = ('1', "hum")
- (e) - : string * string = ("wow", "hum")

5. Quel est le type de la fonction f définie ci-dessous ?

```
let f = function  
  (0,_) -> (0, true)  
  | (_,0) -> (0, false)  
  | (x,y) -> (x/y, true) ;;
```

- (a) int -> int -> int -> bool
- ☒ (b) int * int -> int * bool
- (c) int * int -> int -> bool
- (d) int -> int -> int * bool

6. Quel est le type de la fonction f définie ci-dessous ?

```
let f = function
  ((1,true),_) -> failwith "incorrect"
  | (x, "aa")   -> let (a,b) = x in a ;;
```

- (a) `int * bool * string -> int`
- ☒ (b) `(int * bool) * string -> int`
- (c) `(int * bool) * string -> string`
- (d) `(int * bool) * string -> 'a`
- (e) La fonction est incorrecte.

7. Quel est le type de la fonction f définie ci-dessous ?

```
let f = function
  "et"      -> (function (v1,v2) -> v1 && v2)
  | "ou"     -> (function (v1,v2) -> v1 || v2)
  | "oue"    -> (function (v1,v2) -> (v1 <> v2))
  | "implique" -> (function (v1,v2) -> not v1 || v2)
  | _        -> (failwith "unknown operator") ;;
```

- (a) `string * bool * bool -> bool`
- (b) `string -> bool -> bool -> bool`
- ☒ (c) `string -> bool * bool -> bool`
- (d) `string -> bool -> bool`
- (e) La fonction est incorrecte.

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

```
let h x y = match (x, y) with
  (x, x) -> true
  | _     -> false ;;
```

- (a) `val h : 'a -> 'a -> bool = <fun>`
- (b) `val h : 'a * 'a -> bool = <fun>`
- (c) `val h : 'a -> 'b -> bool = <fun>`
- ☒ (d) Une erreur.

9. Quel est le résultat de l'évaluation de la fonction suivante ?

```
let h c = match c with
  (x, y) when x = y -> true
  | _                -> false ;;
```

- (a) `val h : 'a -> 'a -> bool = <fun>`
- ☒ (b) `val h : 'a * 'a -> bool = <fun>`
- (c) `val h : 'a -> 'b -> bool = <fun>`
- (d) Une erreur.

10. Soient x et y deux valeurs entières définies. Quelles expressions sont équivalentes à l'expression suivante ?

```
let y = x in y + 1 ;;
```

- ☒ (a) `match x with y -> y + 1`
- (b) `let x = y in x + 1`
- (c) `let x = y in y + 1`
- (d) `y + 1`
- ☒ (e) `x + 1`