



## Réponses 3 (Combien ? - 4 points)

## 1. Spécifications :

La fonction `how_many` prend en paramètre une fonction  $f$  ainsi qu'une liste  $[a_1, a_2, \dots, a_n]$ . Elle retourne le nombre de valeurs  $a_i$  telles que  $f(a_i)$  soit vrai.

Fonction CAML :

let how\_many f l:  
 let rec how l m = match l with  
 | [] -> m  
 | e::q when f e -> how q (m+1)  
 | e::q -> how q m  
 in how l 0;;  
 - val how\_many : ('a -> bool) -> 'a list -> int <fun>

## 2. Spécifications :

La fonction `count_multiples n l` retourne le nombre de multiples de  $n$  dans la liste  $l$ .

Fonction CAML :

let mbrat l m = how\_many (function x -> x mod n = 0) l;;  
 - val mbrat : int list -> int -> int <fun>

Réponses 4 (exists2 - 6 points)

1. Spécifications :

exists2 p [a<sub>1</sub>; a<sub>2</sub>; ...; a<sub>n</sub>] [b<sub>1</sub>; b<sub>2</sub>; ...; b<sub>n</sub>] teste s'il existe au moins un couple (a<sub>i</sub>, b<sub>i</sub>) tel que p a<sub>i</sub> b<sub>i</sub> est vrai. Elle déclenche une exception si les deux listes sont de longueurs différentes (si aucun couple (a<sub>i</sub>, b<sub>i</sub>) tel que p a<sub>i</sub> b<sub>i</sub> est vrai n'a été trouvé...).

Fonction CAML :

```
let exists2 p l1 l2 =
  let rec exist l1 l2 = match (l1, l2) with
  | ([], []) -> false
  | ([], _) | (_, []) -> invalid_arg "listes différentes"
  | (e1::q1, e2::q2) when p e1 e2 -> true
  | (e1::q1, e2::q2) -> exist q1 q2
  in exist l1 l2 ;;

- val exists2 : ('a -> 'b -> bool) -> 'a list -> 'b list -> bool = <fun>
```

2. Spécifications :

La fonction identical teste si deux listes sont identiques. Elle déclenche une exception si les deux listes sont de longueurs différentes.

Fonction CAML :

```
let identical l1 l2 = (exists2 / fonction x -> fonction y -> x == y) l1 l2 = false ;;

- val identical : 'a list -> 'a list -> bool = <fun>
```