

I	0
II	0
III	3
IV	5.5

Axiomes de l'opération *card* (extension au type *ensemble*) :

### 1. Spécifications :

## 2. Fonction CAML :



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\_many\_rec f l x = match l with  
 | [] → x  
 | e :: l when f e → how\_many\_rec f l (x+1)  
 | e :: l → how\_many\_rec f l x  
 in how\_many\_rec f 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 count\_multiples n l =  
 how\_many (function x → x mod n = 0) l  
 val count\_multiples : int → int list → int = <fun>



Réponses 4 (exists2 - 6 points)

1. Spécifications :

exists2 p  $[a_1; a_2; \dots; a_n]$   $[b_1; b_2; \dots; b_n]$  teste s'il existe au moins un couple  $(a_i, b_i)$  tel que  $p a_i b_i$  est vrai. Elle déclenche une exception si les deux listes sont de longueurs différentes (si aucun couple  $(a_i, b_i)$  tel que  $p a_i b_i$  est vrai n'a été trouvé...).

Fonction CAML :

let rec exists2 p l1 l2 = match l1 l2 with	
[], []	→ false
l1, []   [], l2	→ invalid arg "Taille différente"
a :: l1, b :: l2 when p a b	→ true
a :: l1, b :: l2	→ exists2 p l1 l2
val exists2 : (α → β → 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 (=) l1 l2	
val identical : 'a list → 'b list → (α → β → bool) → 'a list → 'b list → bool = <fun>	