

Nom	DAVID
Prénom	Clément
Groupe	B2

Note	17,5
------	------

## Algorithmique

S1

Contrôle n° 1 (C1)

24 Oct. 2016

Feuilles de réponses

I	3,5
II	4
III	4
IV	6

Réponses 1 (Types Abstraits : Ensemble (Nombre d'éléments) - 5 points)

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

$$\text{card}(\text{ensemble-vide}) = 0 \quad \checkmark$$

$$\text{card}(\text{ajouter}(y, e)) = \text{card}(e) + 1 \quad \checkmark$$

$$(y \in e = \text{true}) \Rightarrow \text{card}(\text{supprimer}(y, e)) = \text{card}(e) - 1 \quad \checkmark$$

$$(y \in e = \text{false}) \Rightarrow \text{card}(\text{supprimer}(y, e)) = \text{card}(e) \quad \checkmark$$

$$\text{card}(\text{supprimer}(y, (\text{ajouter}(y, e)))) = \text{card}(e)$$

Réponses 2 (Type abstrait *Liste récursive* → liste CAML - 5 points)

1. Spécifications : - Elle prend en paramètre un élément et une liste

- Elle ajoute l'élément à sa place dans une liste triée croissante

- Elle retourne la liste avec son élément ajouté

2. Fonction CAML :

let rec mystery x l = match l with

| [] -> [x]

| e::q when x <= e -> x::e::q or x::l

| e::q -> e::(mystery x q);;

val mystery : 'a -> 'a list -> 'a list = <fun>

## 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>
```