

Nom	
Prénom	
Groupe	

Note	
------	--

Algorithmique CAML

INFO-SUP S1 EPITA

Examen B1

30 octobre 2024

Remarques (à lire !) :

- ☐ Vous devez répondre directement **sur ce sujet**.
 - Répondez dans les espaces prévus, **les réponses en dehors ne seront pas corrigées**.
 - Aucune réponse au crayon de papier ou au stylo rouge ne sera corrigée.
- ☐ **CAML :**
 - Tout code CAML non indenté ne sera pas corrigé.
 - En l'absence d'indication dans l'énoncé, les seules fonctions que vous pouvez utiliser sont **failwith** et **invalid_arg** (aucune autre fonction prédéfinie de CAML).
 - Une seule version doit être présentée pour chaque fonction à écrire.
 - Vos fonctions doivent impérativement respecter les exemples d'applications donnés.
 - Vous pouvez également écrire vos propres fonctions, dans ce cas elles doivent être documentées (on doit savoir ce qu'elles font).

Dans tous les cas, la dernière fonction écrite doit être celle qui répond à la question.

 - **Tout code CAML doit être suivi du résultat de son évaluation (fait partie de la note) : la réponse de CAML .**
- ☐ La présentation est notée.
- ☐ Durée : 1h30

1. Écrire la fonction `insert_x x order lst` qui prend en paramètres :
 - un élément `x`
 - une fonction à deux paramètres `order` retournant une valeur booléenne
 - une liste `lst` triée selon l'ordre induit par la fonction `order`et qui insère l'élément `x` dans la liste `lst` triée selon l'ordre induit par la fonction `order`.

Fonction CAML :

[illegible]

- La fonction **doit** utiliser la fonction précédente `insert_x` x order lst.

Fonction CAML :

[illegible]

Exercice 3 (Mystery – 4 points)

Soit la fonction `mystery` définie ci-dessous

```
let mystery lst =  
  let rec tmp lst acc switch = match (lst, switch) with  
    ([], true) -> tmp [] acc false  
    | ([], false) -> (match acc with  
      [] -> []  
      | h::acc -> h :: tmp lst acc true)  
    | (h::t, true) -> h :: tmp t (h::acc) true  
  in  
    tmp lst [] true;;
```

1. Donner le type de la fonction `mystery`.

2. Donner les résultats des évaluations successives des phrases suivantes.

(a) `# mystery [1; 2; 8; 2; 1];;`

(b) `# mystery ["a"; "b"; "c"; "d"; "e"; "f"];;`
