Nom	
Prénom	
Groupe	
Prof TD	

Note	/	15
------	---	----

Algorithmique Contrôle 1 - Partie 2

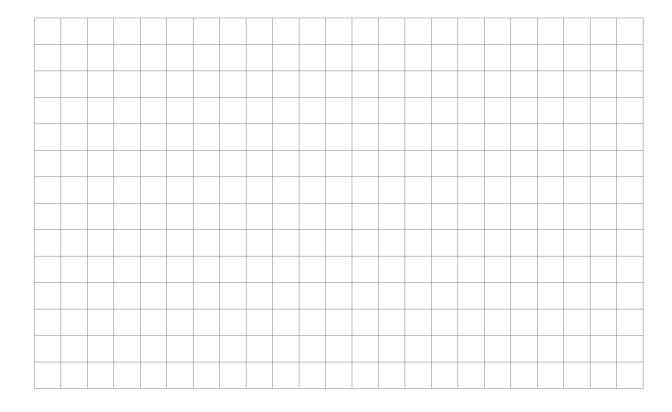
INFO-SUP S1 EPITA

9 Nov. 2020 - 8:30

$\hfill\Box$ Ceci est la partie 2 de l'épreuve - Vous devez rendre les deux parties !
 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 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 son failwith et invalid_arg (aucune autre fonction prédéfinie de CAML). Tout code CAML doit être suivi du résultat son évaluation : la réponse de CAML.
$\hfill\Box$ La présentation est notée.

Exercice 2 (Suppression – 4 points)

Écrire la fonction delete x list qui supprime de la liste list triée en ordre croissant la première occurrence de la valeur x (si elle est présente).



Exercice 3 (Insertion à la $i^{\hat{e}me}$ place – 5 points)

Écrire la fonction insert_nth x i list qui insère la valeur x à la $i^{\grave{e}me}$ place dans la liste list. La fonction devra déclencher une exception Invalid_argument si i est négatif ou nul, ou une exception Failure si la liste est trop courte.

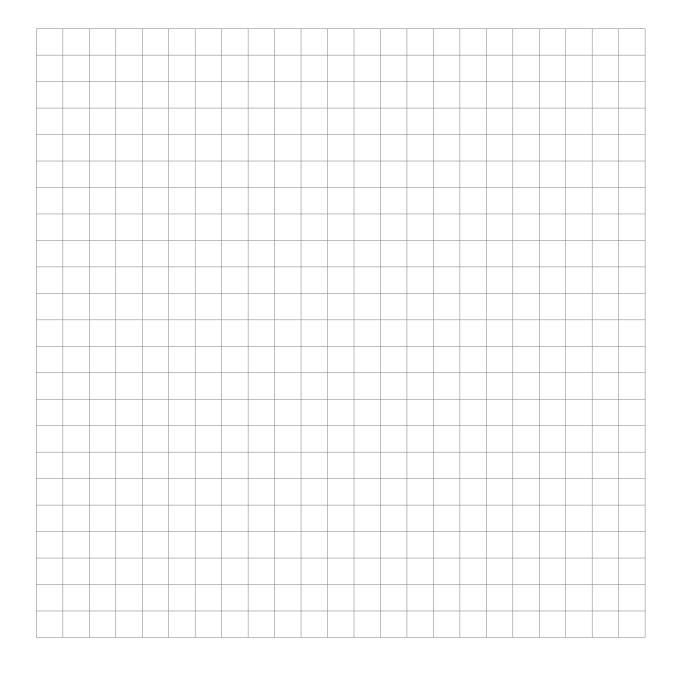
$Exemples\ d'application:$

```
# insert_nth 0 5 [1; 2; 3; 4; 5; 6; 7; 8; 9];;
- : int list = [1; 2; 3; 4; 0; 5; 6; 7; 8; 9]

# insert_nth 0 10 [1; 2; 3; 4; 5; 6; 7; 8; 9];;
- : int list = [1; 2; 3; 4; 5; 6; 7; 8; 9; 0]

# insert_nth 0 12 [1; 2; 3; 4; 5; 6; 7; 8; 9];;
Exception: Failure "out of bound".

# insert_nth 0 (-2) [1; 2; 3; 4; 5; 6; 7; 8; 9];;
Exception: Invalid_arg "negative rank".
```

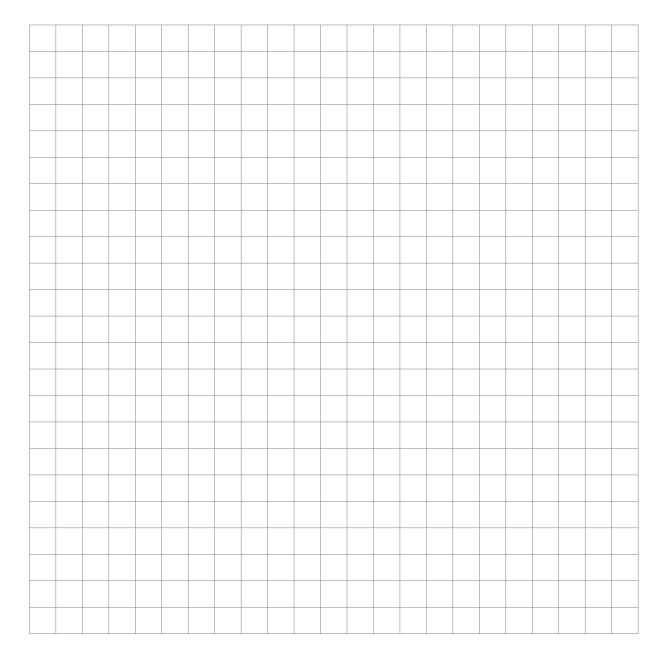


Exercice 4 (Recherche double – 4 points)

Écrire la fonction $\mathtt{search_both}$ list a b qui vérifie si les deux valeurs a et b distinctes sont présentes dans la liste list.

 $Exemples\ d'application:$

```
# search_both [12; 5; -4; 0; 7; 21; 3] 5 0;;
- : bool = true
# search_both [12; 5; -4; 0; 7; 21; 3] 21 (-4);;
- : bool = true
# search_both [12; 5; -4; 0; 7; 21; 3] 0 42;;
- : bool = false
```



Exercice 5 (Mystery - 2 points)

Soit la fonction go définie ci-dessous

```
let go = function
   [] -> []
   | e::list ->
   let rec what x = function
        [] -> []
        | e::list -> (e * x)::(what e list)
   in
   what e list;;
```

Donner les résultats des évaluations successives des phrases suivantes.

```
# go [1; 1; 1; 1; 1] ;;

# go [42] ;;

# go [1; 2; 3; 4; 5] ;;

# go [2; 21; 2; 21; 2; 21] ;;
```



