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.

Exercice 1 (Vérification liste de listes – 7 points)

Écrire la fonction check_list k 1st qui prend en paramètres :

- un entier k
- une liste de listes 1st

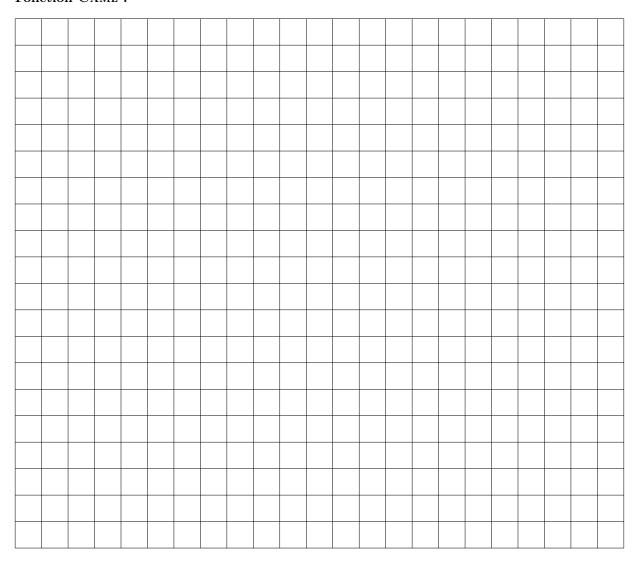
et qui vérifie si la liste de listes 1st contient uniquement des listes de longueur $\leq k$.

La fonction devra déclencher une exception $Invalid_argument$ si le paramètre k est < 0.

Rappel : les seules fonctions que vous pouvez utiliser sont failwith et invalid_arg (aucune autre fonction prédéfinie de CAML).

```
# check_list 2 [[1; 2]; [4]; [5; 6]];;
- : bool = true
# check_list 3 [[1; 2; 3]; [4; 5; 6; 7]; [8; 9; 10]];;
- : bool = false
# check_list 1 [[]; [2]; [3]];;
- : bool = true
# check_list (-1) [[1]; [2]; []];;
Exception: Invalid_argument "check_list: k < 0".</pre>
```

Fonction Caml:



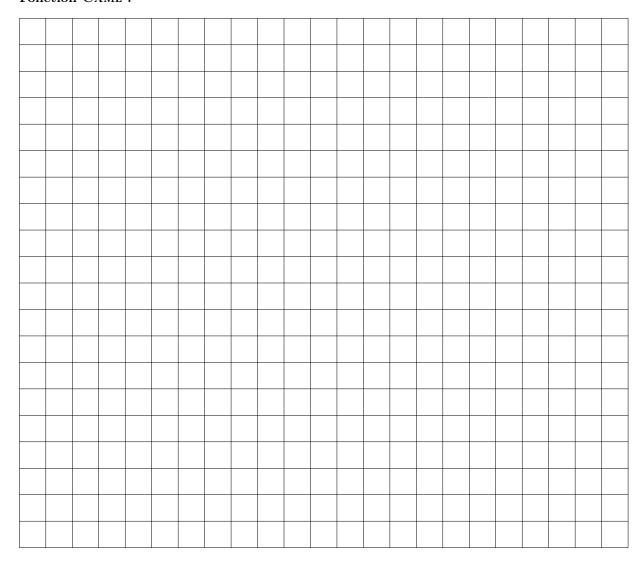
Exercice 2 (Tri insertion – 9 points)

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

et qui insère l'élément x dans la liste 1st triée selon l'ordre induit par la fonction order.

```
# insert_x 3 (function x -> function y -> x < y) [];;
- : int list = [3]
# insert_x 7 (function x -> function y -> x < y) [1; 3; 5; 9];;
- : int list = [1; 3; 5; 7; 9]
# insert_x 5 (function x -> function y -> x > y) [10; 8; 6; 4; 2];;
- : int list = [10; 8; 6; 5; 4; 2]
# insert_x 'c' (function x -> function y -> x < y) ['d'; 'e'];;
- : char list = ['c'; 'd'; 'e']
# insert_x 'c' (function x -> function y -> x > y) ['e'];;
- : char list = ['e'; 'c']
```

Fonction Caml:

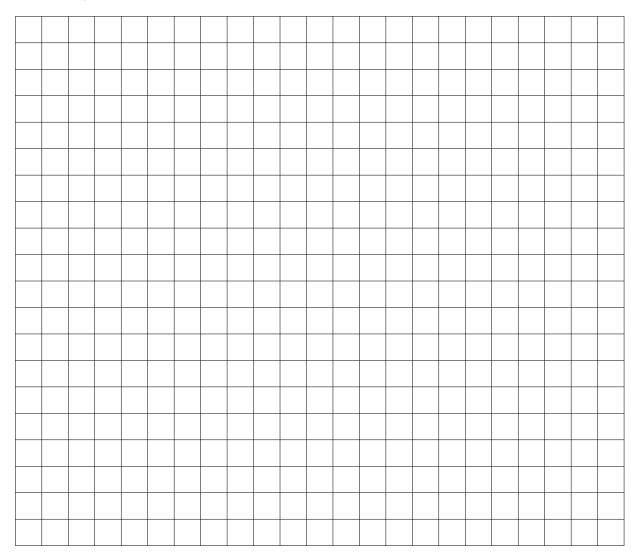


- 2. Écrire la fonction insertion_sort order 1st qui prend en paramètres :
 - une liste 1st
 - une fonction à deux paramètres **order** retournant une valeur booléenne et qui trie la liste 1st selon l'ordre induit par la fonction **order**.

La fonction doit utiliser la fonction précédente insert_x x order lst.

```
# insertion_sort (function x -> function y -> x > y) [];;
- : 'a list = []
# insertion_sort (function x -> function y -> x < y) [5; 3; 8; 1; 4];;
- : int list = [1; 3; 4; 5; 8]
# insertion_sort (function x -> function y -> x > y) [10; 20; 15; 5; 30];;
- : int list = [30; 20; 15; 10; 5]
# insertion_sort (function x -> function y -> x < y) ['d'; 'a'; 'c'; 'b'; 'e'];;
- : char list = ['a'; 'b'; 'c'; 'd'; 'e']
# insertion_sort (function x -> function y -> x < y) [1.1; 3.3; 2.2; 5.5; 4.4];;
- : float list = [1.1; 2.2; 3.3; 4.4; 5.5]</pre>
```

Fonction Caml:



Exercice 3 (Mystery - 4 points)

Soit la fonction mystery définie ci-dessous

1. Donner le type de la fonction mystery.

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

```
(a) \sharp mystery [1; 2; 8; 2; 1];;
```

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