

Interrogation écrite 3

INF 201 — IMA3&4 — 12/04/2023 — 30 minutes

Exercice 1. (/5) La suite de Lucas u associée aux entiers p et q est définie par:

$$u_0(p, q) = 0 \quad u_1(p, q) = 1 \quad \text{et} \quad u_n(p, q) = p u_{n-1}(p, q) - q u_{n-2}(p, q) \quad , n \geq 2$$

Spécifier et réaliser `lucas p q n`:

PROFIL (/1):

EXEMPLE (/1) : `lucas 1 (-1) 3 =`

IMPLÉMENTATION (/3):

Exercice 2. (/6) Dans cet exercice de cours on implémente le tri par insertion de façon **récurive**.

Question 1. (/3) Implémenter une fonction `insérer` qui insère un élément dans une liste triée.

```
val insérer : 'a->'a list -> 'a list = <fun>
```

Question 2. (/3) En déduire une fonction `tri` qui trie une liste.

```
val tri : 'a list -> 'a list = <fun>
```

Exercice 3. (/6) On veut maintenant réaliser le produit cartésien de deux listes. On rappelle que pour deux ensembles A et B :

$$A \times B = \bigcup_{b \in B} A \times \{b\}$$

Par exemple:

```
cartesian [1;2] [3;4];;
```

```
- : (int * int) list = [(1, 3); (1, 4); (2, 3); (2, 4)]
```

Question 3. (/3) En utilisant le schéma d'ordre supérieur `List.map`, écrire une fonction `add_elem` non récursive qui ajoute un élément à chaque élément d'une liste:

```
val add_elem : 'a->'b list -> ('a*'b) list = <fun>
```

Question 4. (/3) En déduire une fonction `cartesian` qui effectue le produit cartésien de deux listes à l'aide du schéma d'ordre supérieur `List.fold_right`

```
val cartesian : 'a list->'b list -> ('a*'b) list = <fun>
```

Exercice 4. (/7) Dans cet exercice on veut faire la composition d'un nombre arbitraire de fonctions. Pour cela on va écrire une fonction `comp_multi` qui compose les fonctions contenues dans une liste:

Exemple: `comp_multi [f;g;h] = f o g o h`

Pour cela on utilisera l'opérateur de composition vu en TD: `>>`

Question 5. (/1) Rappeler le profil de `>>`

Question 6. (/2) On définit une fonction par:

```
let h = comp_multi [(fun x -> 2*x); (fun x -> x+1); (fun x -> 3*x)];;
```

Donner son profil et calculer `h 1`

Question 7. (/4) Implémenter `comp_multi` à l'aide d'un schéma d'ordre supérieur.