

Université d'Orléans	L2 Informatique
Programmation fonctionnelle	2022-2023

## TD n°4

### Exercice 1

Pour chaque paire de motif et d'expression, indiquer si le motif filtre l'expression, et si c'est le cas, les liaisons créées.

x	42
x	(42, true)
(n, b)	(42, true)
(n, b, s)	(42, true)
(n, b, _)	(42, true)
[]	[42]
[n]	[42]
[_]	[42]
_::[]	[42]
n::[]	[42]
n::l	[42]
_::l	[42]
x::y::[]	[1.1; 2.2; 3.3]
x::y::[]	[0.5; 2.5]
(x,y)::_	([1; 2], [3; 4])
(x,y)::_	[(1,2); (3,4)]

### Exercice 2

Sans l'utiliser, donner la réponse que donnerait la boucle interactive pour chacune des phrases suivantes, en supposant que celles-ci sont évaluées à la suite. Préciser à chaque fois s'il s'agit d'une expression ou d'une définition. Dans le cas d'une définition, donner les liaisons globales obtenues.

```

1  let (answer_num, answer_string) = (42, "Forty-Two");;
2  let (answer_num', _) = (42, "Forty-Two");;
3  let (x, (y, z)) = (1, (2, 3));;
4  let middle = match [1; 2; 3] with
5      | [] -> []
6      | [_ ; n; _] -> [ n ]
7      | _ -> [];;
8  let rec combine = fun (l1,l2) ->
9      match (l1, l2) with
10     | ([], _) -> []
11     | (_, []) -> []
12     | (h1::t1, h2::t2) -> (h1, h2)::(combine(t1, t2));;
13  combine([1;2;3], [1.; 2.; 3.]);;
14  combine([], ["Bonjour"; "tout"; "le"; "monde"]);;
15  let f = fun n -> match n with
16      | 1 -> "Red"
17      | 2 -> "Green"
18      | 3 -> "Blue"
19      | _ -> "Not a color";;
20  f answer_num;;
21  f x;;

```

```

22     f z;;
23     f y;;
24     f (-1) ;;

```

### Exercice 3

Sans l'utiliser, donner la réponse que donnerait la boucle interactive pour chacune des phrases suivantes, en supposant que celles-ci sont évaluées à la suite. Préciser à chaque fois s'il s'agit d'une expression ou d'une définition. Dans le cas d'une définition, donner les liaisons globales obtenues.

```

1   List.map ((+) 2) [1; 2; 4; 8];;
2   List.filter ((<=) 4) [4; 2; 6; 3];;
3   List.map (List.map (( * ) 2)) [ [1; 2]; [4; 2]; [0; 1] ];;
4   fun s -> fun r -> s^" = "^(string_of_int r);;
5   let string_val = fun s -> fun r -> s^" = "^(string_of_int r);;
6   let i = string_val "x";;
7   i 2;;
8   string_val "answer" 42;;

```

### Exercice 4

#### Question 1

En utilisant le filtrage de motifs et la récursion, écrire une fonction `last` : `'a list -> 'a` qui renvoie le dernier élément d'une liste si celle-ci n'est pas vide.

#### Question 2

En utilisant le filtrage de motifs et la récursion, écrire une fonction `without_last` : `'a list -> 'a list` qui renvoie une liste privée de son dernier élément. La liste argument ne doit pas être vide.

#### Question 3

Sans utiliser les fonctions des questions 1 et 2, mais en utilisant le filtrage de motifs et la récursion, écrire une fonction `cut_last` : `'a list -> 'a * 'a list` qui prend en argument une liste non-vide et renvoie un couple formé de son dernier élément, et de la liste privée de ce dernier élément.

La fonction `cut_last` est telle que pour toute liste non vide `l`, l'égalité suivante est vraie :

```
let (last, without_last) = cut_last l in (l = without_last @ [ last ])
```

En utilisant `cut_last`, donner de nouvelles versions des fonctions `last` et `without_last`.

#### Question 4

Ecrire une fonction `shift_right` qui décale une liste d'éléments circulairement vers la droite (c'est-à-dire que le dernier élément devient le premier élément).

Comparer le nombre d'appels nécessaires pour évaluer une application de `shift_left` et une application de `shift_right`.