

Correction du partiel

22 octobre 2002

Exercice 1

Exercice 1

```
1 # let rec f =
2     fun m n -> if n=0 then m else 2 + f m (n-1) ;;
3 val f : int -> int -> int = <fun>
4
5 (* elle calcule (m,n) -> m + 2*n *)
6 let f2 = fun m n -> m + 2*n ;;
```

Exercice 2

Exercice 2

```
1 # sqrt (float_of_int 5) ;;
2 - : float = 2.2360679775
3
4 # let c = 0 in (fun c -> if c='c' then "oui" else "non");;
5 - : char -> string = <fun>
6
7 # fun c -> fun y -> c ^ (y c) ;;
8 - : string -> (string -> string) -> string = <fun>
9
10 # let g = List.hd in fun l -> (g l)::l ;;
11 - : 'a list -> 'a list = <fun>
```

Exercice 3

Exercice 3

```
1 # let rec nettoyer l = match l with
2     [] -> []
3     | e::m -> if (e="") then nettoyer m else e::(nettoyer m) ;;
```

```
4
5
6 # let rec nettoyer = fun l ->
7   if l=[] then [] else
8     let h= List.hd l in
9       let t= nettoyer (List.tl l) in
10        if h="" then t else (h::t) ;;
11
12 # let nettoyer = List.filter (fun s -> s<>"") ;;
```

Exercice 4

Exercice 4

```
1 # let longueurs =
2   fun l -> List.map (fun s -> (s,String.length s)) (nettoyer l);;
```

Exercice 5

Exercice 5

```
1 # let rec nbdiv n a = if (a<=1) then 1
2   else (if (n mod a = 0) then 1 else 0) + nbdiv n (a-1) ;;
3
4 # let trois_fact n = (nbdiv n n = 3);;
```
