corrigé TD n°8

Pour rendre les programmes plus lisibles, on peut définir un type “big_number” et forcer le typage des fonctions que nous écrivons:

```
#type big_number == int list;;
Le type big_number est défini.

let somme (nombre1:big_number) (nombre2 : big_number) =
let rec som = fun
 [] [] 0 -> []
|[] [] n -> [n]
|[] (y::ys) n -> let inter = y+n in
                 (inter mod 100)::(som [] ys (inter/100))
|(x::xs) [] n -> let inter = x+n in
                 (inter mod 100)::(som xs [] (inter/100))
|(x::xs) (y::ys) n -> let inter = x+y+n in
                 (inter mod 100)::(som xs ys (inter/100)) in
(som nombre1 nombre2 0 : big_number);;
somme : big_number -> big_number -> big_number = <fun>
#somme [45;23][55;76;99;99;99];;
- : big_number = [0; 0; 0; 0; 0; 1]

#let rec affiche = function
 [] -> failwith "erreur"
| [x] -> print_int x
| x::xs when x > 9 -> affiche xs; print_int x
| x::xs when  x > 0 -> affiche xs; print_int 0; print_int x
| x::xs -> affiche xs; print_int 0; print_int 0;;

affiche : int list -> unit = <fun>
#affiche(somme [45;23][55;76;99;99;99]);;
10000000000- : unit = ()

#let rec string_of_big_int = function
 [] -> failwith "erreur"
| [x] -> string_of_int x
| x::xs when x > 9 -> (string_of_big_int xs)^( string_of_int x)
| x::xs when  x > 0 -> (string_of_big_int xs)^"0"^(string_of_int x)
| x::xs -> (string_of_big_int xs)^"00";;

string_of_big_int : int list -> string = <fun>
#string_of_big_int([55;76;99;99;99]);;
- : string = "9999997655"

#let rec big_int_of_int n =
let rec transforme = function
  n when n <100 -> [n]
 | n -> (n mod 100)::(transforme (n/100))
in (transforme n : big_number);;
big_int_of_int : int -> big_number = <fun>
#big_int_of_int 20123542;;
- : big_number = [42; 35; 12; 20]

#let produit (l1 : big_number) (l2 : big_number) =
let rec prod x = fun
  []  r -> if r = 0 then []
                 else [r]
| (y::ys) r -> let p = x*y+r in
           (p mod 100)::(prod x ys (p/100))
```

```
in
let rec produit_rec = fun
  [] l2 -> []
| (x::xs) l2 -> somme (prod x l2 0) (0::(produit_rec xs l2))
in
   (produit_rec l1 l2 : big_number);;
produit : big_number -> big_number -> big_number = <fun>
#affiche(produit (big_int_of_int 124124) (big_int_of_int 125458));;
15572348792- : unit = ()

#let rec fact = function
 n when n<0 -> failwith "Erreur"
| 0 -> ([1] : big_number)
| n -> produit (big_int_of_int n) (fact (n-1));;

fact : int -> big_number = <fun>
#affiche(fact 123)
;;
12146304367025329675766243241881295855454217088483382315328918161829235892362167
66883115696061264020217073583522129404778259109157041165147218602951990626164673
07339074198149529600000000000000000000000000000- : unit = ()
#affiche(fact 23)
;;
25852016738884976640000- : unit = ()
```