

TD7 - Introduction à Lustre

Ex1. Pre ou pas Pre

```
node udtqc (x : int) returns (y : int);
let
  y = 1 -> pre (2 -> pre (3 -> pre (4 -> 5)));
tel;
```

Commande	Output
5	5 5 5 5 5 5 5 5
4	4 4 4 4 4 4 4 4
4 → 5	4 5 5 5 5 5 5 5
pre(4 → 5)	∅ 4 5 5 5 5 5 5
3	3 3 3 3 3 3 3 3
3 → pre(4 → 5)	3 4 5 5 5 5 5 5

Valeur finale: 1 2 3 4 5 5 5 5 ..

Ex2. Compteur dans un musée

```
node ces (in, out:bool) returns (compteur : int);
let
  var cin, cout : int,
  cin = if in then 1 else 0;
  cout = if out then -1 else 0;
  compteur = (cin + cout + (0 -> pre compteur));
tel;
```

Ex3. Fibonacci

Q1.

```
# Hack: nil par défaut dans un + a la valeur 0
node fib (x : bool) returns (y : int);
  let y = 1 -> (pre y) + (pre (pre y));
tel;
```

```
# Mieux
node fib (x : bool) returns (y : int);
  let y = 1 -> (pre 1 -> y + pre y);
tel;

# Équivalent:
node fib (x : bool) returns (y : int);
  let y = 1 -> (y + (0 -> pre y));
tel;
```

Q2.

```
node fib (max : int) returns (y : int);
let y =
  if (1 -> (pre y)) > max then
    1 -> (y + (0 -> pre y))
  else
    pre y;
tel;
```

Q3.

Ex4. When et Current

Q1.

X	1	2	3	4	5	6	7	8	9
Y	true	false	true	false	true	false	true	false	true
Z	true	true	false	false	true	false	false	true	false
A = X when Z	1	2			5			8	
B = Y when Z	true	false			true			false	
C = A when B	1				5				
current(C)	1	1			5			5	

Q2.

```
node modulo_n (n : int) returns (nb, max_mul : int);
let
  inc = 0 -> pre inc;
```

```
(nb, max_mul) = (inc, current(inc when ((inc mod n) = 0)));  
tel;
```