# Corrigé feuille d'exercices n°11 de l'option d'informatique.

## Autour des nombres premiers
## Exercices n°1, 2 et 3

```
let divise a  b = (b mod a)=0;; (* divise : int -> int -> bool = <fun> *)

let rec trouve_diviseur n diviseur =
  match n with
  n when  carre diviseur > n -> n
| n when n mod diviseur = 0 -> diviseur
| n  -> trouve_diviseur n (diviseur+1);;
(* trouve_diviseur : int -> int -> int = <fun> *)

let suivant = function
  2 -> 3
| n when n modulo 3 = 1 -> n+4
| n -> n +2;;

let rec trouve_diviseur n diviseur =
  match n with
  n when  carre diviseur > n -> n
| n when n mod diviseur = 0 -> diviseur
| n  -> trouve_diviseur n (suivant diviseur);;
(* trouve_diviseur : int -> int -> int = <fun> *)


let premier n = n = trouve_diviseur n 2;;
(* premier : int -> bool = <fun> *)
```

## Exercice n° 4 : test de Fermat pour les nombres premiers

```
let est_pair n = (n mod 2 = 0);;

let carre x = x * x;;

let rec expmod b e m = match e with
  0 -> 1
| e when (est_pair e) -> carre (expmod b ( e / 2) m) modulo m
| e -> (b *  expmod b (e-1) m) modulo m;;

let test_fermat n =
   let a = 2 + random__int (n-2) in
       expmod a n n = a;;
```

```
13
14  let rec premier_rapide fois n = match fois with
15    0 -> true
16  | fois when test_fermat n -> premier_rapide (fois - 1) n
17  | _ -> false;;
18
19  premier_rapide 10 919;;
20
21  let rec PGCD a = function
22    0 -> a
23  | b -> PGCD b (a mod b);;
24
25  let est_pair n = (n mod 2 = 0);;
```

**Exercice n° 5 : nombre premier par le test de Solovay-strassen**

On réécrit de la fonction "mod" qui ne marche pas en Caml pour les nombres négatifs

```
1   let modulo a b =
2       match a - (a/b)*b with
3     x when x > 0 -> x
4   | x -> abs(b) + x;;
5
6   #infix "modulo";;
7
8   let rec exp_rapide a = function
9      0 -> 1
10  | n when n mod 2 = 0 -> carre (exp_rapide a (n / 2))
11  | n -> a * (exp_rapide a (n-1));;
12  (* exp_rapide : int -> int -> int = <fun> *)
13
14  let rec J a n = match a with
15     1 -> 1
16  | a when est_pair a -> J (a/2) n * (exp_rapide (-1) ((carre n - 1) / 8))
17  | a -> J (n modulo a) a * (exp_rapide (-1) ((a-1)*(n-1) / 4));;
18
19  let rec estpremier_bis fois n =
20     let a = 2 + random__int (n-2) in
21         match (fois,PGCD a n) with
22       (0,1) -> (J a n modulo n - (expmod a ((n-1)/2) n)) = 0
23    | (k,1) -> if (J a n modulo n = expmod a ((n-1)/2) n) then estpremier_bis (k-1) n else f
24    | _ -> false;;
25
26  for i = 3 to 100 do
27        if estpremier_bis 10 i then begin print_int i; print_newline() end; done;;
```