

DM2 - retour sur trace - corrigé

7.

```
let est_connu c = not(c=I);;
```

8.

```
let get p z = p.(z/n).(z mod n);;
```

9.

```
let set p z c = let p0=presolution_init() in for x=0 to m-1 do for y=0 to n-1 do
  p0.(x).(y) <- p.(x).(y) done done;
  p.(z/n).(z mod n) <- c; p0;;
```

10. Un objet immuable est un objet qui ne peut être modifié après sa définition. Un des avantages est qu'il ne peut y avoir de partage de données avec d'autres objets donc d'altération ou de modification inopinée de sa valeur.

11.

```
let est_complet_lig p x = let rec aux i = match i with
  | n -> true
  | _ -> est_connu (get p (x*n+i)) && aux (i+1) in aux 0;;
```

12.

```
let trace_lig p x = let rec aux liste c i =
  if i=-1 then if c=0 then liste else c::liste else match get p (x*n+i) with
  | N -> aux liste (c+1) (i-1)
  | B -> if c=0 then aux liste c (i-1) else aux (c::liste) 0 (i-1) in aux [] 0 (m-1);;
```

13.

```
let est_admissible h p = let rec aux1 x = match x with
  | m -> true
  | x when est_complet_lig p x -> (trace_lig p x = h.ind_lig.(x) ) && aux (x+1)
  | _ -> aux (x+1)
and aux2 x = match w with
  | n -> true
  | w when est_complet_col p x -> (trace_col p x = h.ind_col.(x) ) && aux (x+1)
  in (aux1 0) && (aux2 0);;
```

14.

```
let etend_trivial h p z c = let pp=set p z c in if ((get p z = I)|| (get p z =c)) && (est_admissible h pp) then Some pp else None;;
```

15.

```
let resout h ext = let p0=presolution_init() in
  let rec explore p z = if z=n*m then Some p else match ext h p z B with
    | Some q -> match explore q (z+1) with
      | Some r -> Some r
      | None -> match ext h p z N with
        | Some q -> match explore q (z+1) with
          | Some r -> Some r
          | None -> None
        | None -> None
    | None -> match ext h p z N with
      | Some q -> match explore q (z+1) with
        | Some r -> Some r
        | None -> None
      | None -> None
  in explore p0 0;;
```