Question 1

```
let init n =
   let v = Array.make n (0,0) in
      for i = 0 to (n-1) do
      v.(i) <- (i,0)
      done;
v;;
```

Question 2

```
let rec find p i = match fst p.(i) with
  | j when i = j -> j
  | j            -> let k = find p j in p.(i) <- (k, snd p.(i)) ; k ;;
```

Question 3

```
let union p i j =
  let ci = find p i and cj = find p j in
  match snd p.(ci), snd p.(cj) with
    | _ when ci = cj      -> ()
    | ri, rj when ri < rj -> p.(ci) <- (cj, snd p.(ci))
    | ri, rj when ri > rj -> p.(cj) <- (ci, snd p.(cj))
    | ri, rj              -> p.(ci) <- (cj, snd p.(ci)) ; p.(cj) <- (cj, snd p.(cj) + 1) ;;
```

Question 4

```
type voisin = int list ;;
type graphe = voisin array ;;

let composantes g =
  let n = Array.length g in
  let p = init n in
  let rec aux i = function
    | []   -> ()
    | j::q -> union p i j ; aux i q in
  for i = 0 to n-1 do aux i g.(i) done ;
  p ;;
```

Question 5

```
open Random

let p = init 1000000 ;;
for k = 1 to 1000000 do
  let i = int 1000000 and j = int 1000000
  in union p i j done ;;

let hauteur p i =
  let rec aux h = function
    | j when fst p.(j) = j -> h
    | j                    -> aux (h+1) (fst p.(j))
  in aux 0 i ;;

let maxhauteur p =
  let n = Array.length p in
  let rec aux acc = function
    | j when j = n -> acc
    | j            -> aux (max acc (hauteur p j)) (j+1)
  in aux 0 0 ;;

maxhauteur p ;;
```