

```
type dict = Nil | Noeud of char * dict * dict ;;
```

```
let chercher m a =  
  let n = string_length m in  
  let rec aux k = function  
    | Nil -> false  
    | Noeud (`.`, _, _) when k = n -> true  
    | Noeud (_, _, a2) when k = n -> aux k a2  
    | Noeud (c, a1, _) when m.[k] = c -> aux (k+1) a1  
    | Noeud (_, _, a2) -> aux k a2  
  in aux 0 a ;;
```

```
let branche m =  
  let n = string_length m in  
  let rec aux = function  
    | k when k = n -> Noeud (`.`, Nil, Nil)  
    | k -> Noeud (m.[k], aux (k+1), Nil)  
  in aux 0 ;;
```

```
let inserer m a =  
  let n = string_length m in  
  let rec aux k = function  
    | Nil -> branche (sub_string m k (n-k))  
    | Noeud (`.`, a1, a2) as a when k = n -> a  
    | Noeud (c, a1, a2) when k = n -> Noeud (c, a1, aux k a2)  
    | Noeud (c, a1, a2) when m.[k] = c -> Noeud (c, aux (k+1) a1, a2)  
    | Noeud (c, a1, a2) -> Noeud (c, a1, aux k a2)  
  in aux 0 a ;;
```

```
let rec creer = function  
  | [] -> Nil  
  | t::q -> inserer t (creer q) ;;
```

```
let extraire a =  
  let rec aux acc = function  
    | Nil -> []  
    | Noeud (`.`, _, a2) -> acc::(aux acc a2)  
    | Noeud (c, a1, a2) -> (aux (acc ^ (string_of_char c)) a1) @ (aux acc  
a2)  
  in aux "" a ;;
```