1. (a)
```
for i:=1 to longueur(l) do
  write(element(l,i)) ;
```

(b)
```
begin
  for i:=1 to longueur(l) do
    if not double(l,element(l,i))
    then write(element(l,i)) ;
  end.

  function double(l:liste,x:E) : boolean ;
  var i, : integer ;
  begin
    c := 0 ;
    for i:=1 to longueur(l) do
      if element(l,i) = x then c := c + 1 ;

    double := (c > 1) ;
  end ;
```

(c)
```
for i:=1 to longueur(l) do begin
  cour := element(l,i) ;
  j := 1 ;
  while j <= longueur(l) do
    if element(l,j) = cour then supprimer(l,j)
    else j := j + 1 ;
end ;
```

On peut aussi envisager le problème de façon très différente : en une seule passe, on peut compter les occurrences de chaque élément. Ensuite, on n'a besoin que d'une autre passe pour éliminer les doublets.

2. Version itérative

```
for i:=length(x) downto 1 do
  write(x[i]) ;
writeln ;
```

Version récursive

```
procedure miroir(x:string) ;
var l : integer ;
begin
  l := length(x) ;
  if l = 1 then write(x[1])
  else begin
    write(x[l]) ;
    miroir(copy(x,1,l-1)) ;
  end ;
end ;
```

3. Cf. cours.