

```

1. mot_cour:='' ;
   for i:=1 to N (*length(texte)*) do begin
     if texte[i]=' ' then begin
       writeln(mot_cour) ;
       mot_cour := '' ;
     end
     else mot_cour := mot_cour+texte[i] ;
   end ;

```

2. (a) Structure de donnée

```

type chaine = string[22] ;
type mot = record
  s : chaine ;      (* mot *)
  o : integer ;    (* nb occurrences *)
end ;
type liste = record
  t : array[1..1000] of mot ; (* liste *)
  l : integer ;      (* nb de mots *)
end ;
var dico : liste ;

```

On suppose que le nombre effectif de mots dans la liste est indiqué par `dico.l`. Le troisième enregistrement de la liste (par exemple) est référencé par `dico.t[3]`. Le mot correspondant est référencé par `dico.t[3].s`, et son nombre d'occurrences par `dico.t[3].o`. Enfin, la sixième lettre de ce mot peut être référencée par `dico.t[3].s[6]`.

(b) (i) Tri : Par exemple, tri par sélection-échange

```

for i:=1 to dico.l do begin
  PosMin := i ;
  CleMin := dico.t[i].s
  for j := i to dico.l do
    if dico.t[j].s < CleMin then begin
      PosMin := j ;
      CleMin := dico.t[j].s ;
    end ;
  echanger(dico.t[j], dico.t[PosMin]) ;
end ;

```

(ii) Remplissage direct : chaque mot est rangé à sa place dans la structure, ce qui veut dire qu'il faut décaler d'un cran le reste du tableau (et gérer la longueur). C'est la procédure `ranger` qui s'occupe de cela.

```

programme stats ;
type chaine = string[22] ;
type mot = record
  s : chaine ;      (* mot *)
  o : integer ;    (* nb occurrences *)
end ;
type liste = record
  t : array[1..1000] of mot ; (* liste *)
  l : integer ;      (* nb de mots *)
end ;
var dico : liste ;
    i : integer ;
    mot_cour := chaine ;

```

---

```

procedure ranger(m : chaine ; var d : liste) ;
var i,j : integer ;
begin
  i:=1 ;
  while (m > d.t[i].s and i <= d.l) do
    i := i + 1 ;
    (* décalage vers la droite -- à rebours *)
    for j := d.l downto i do
      d.t[j+1].s := d.t[j].s
    d.t[i].s := m ;
    d.l = d.l + 1 ;
  end ;
  (* -----*)
begin (* programme principal *)
  (* On suppose que la variable texte est remplie ici *)
  mot_cour:='' ;
  for i:=1 to length(texte) do begin
    if texte[i]=' ' then begin
      ranger(mot_cour, dico) ;
      mot_cour := ' ' ;
    end
    else mot_cour := mot_cour+texte[i] ;
  end ;
end.

```

```

3. function appartient(x : valeur ; t : tableau; d,f : integer) : boolean ;
var i : integer ;
begin
  i := d ;
  while (i <= f and x <> t[i])
    i := i + 1 ;
  if i <= f then appartient := true
  else appartient := false ;
end ;

```

```

function appartient(x : valeur ; t : tableau; d,f : integer) : boolean ;
begin
  if x = t[d] then appartient := true
  else if (d=f) then appartient := false
    else appartient := appartient(x,t,d+1,f) ;
end ;

```

```

4. for i:=1 to 10 do begin
  writeln('Entrer un caractère') ;
  readln(x) ;
  deja_la := false ;
  for j:= 1 to longueur(liste)
    if x = element(j,liste) then
      deja_la := true
  if not deja_la then
    inserer(x,longueur(liste)+1,liste) ;
end ;

```

Variante avec localiser

```

for i:=1 to 10 do begin
  writeln('Entrer un caractère') ;
  readln(x) ;
  if localiser(x,liste) = 0 then
    inserer(x,longueur(liste)+1,liste) ;
end ;

```