

Algorithme 1 : Exercice 18

```

Const N = 1000;
Var
text : chaine [1..N]de caractere;
mot : chaine[1..15]de caractere;
nb,i,j:entier;
fonction mot_palindrome(text : chaine [1..N]de caractere):entier;
Var nb_palin : entier;
début
i ← 1; j ← 1; nb_palin ← 0;
tant que i ≤ longueur(text) faire
  tant que i ≤ longueur(text) et text[i]=' ' ou text[i]='.' faire
    i ← i+1;
  j ← i + 1;
  tant que j ≤ longueur(text) et (text[j] ≥ 'a' et text[j] ≤ 'z' ou text[j] ≥ 'A' et text[j] ≤ 'Z' ) faire
    j ← j + 1;
  j ← j - 1;
  tant que i ≤ j et text[i] = text[j] faire
    i ← i + 1;
    j ← j - 1;
  si i > j alors nb_palin ← nb_palin + 1;
  tant que i ≤ longueur(text) et (text[i] ≥ 'a' et text[i] ≤ 'z' ou text[i] ≥ 'A' et text[i] ≤ 'Z' ) faire
    i ← i+1;
mot_palindrome ← nb_palin;
fin;
procédure optimise_blancs(text : chaine [1..N]de caractere);
Var s : entier;
début
s ← 0; j ← 1;
pour i ← 1 à longueur(text) faire
  text[j] ← text[i];
  si text[j] ≠ ' ' alors
    j ← j + 1;
    s ← 0;
  sinon si text[j] = ' ' et s=0 alors
    j ← j + 1;
    s ← s + 1;
fin;
fonction occurrence_mot(text : chaine [1..N]de caractere,mot: chaine[1..15]de caractere): entier;
Var comp:entier;
début
i ← 1; comp ← 0;
tant que i ≤ longueur(text) faire
  j ← 1;
  tant que i ≤ longueur(text) et j ≤ longueur(mot) et text[i] = mot[j] faire
    i ← i + 1;
    j ← j + 1;
  si j > longueur(mot) et text[i]=' ' ou text[i]=' \0' alors
    comp ← comp + 1;
  i ← i + 1;
occurrence_mot ← comp;
fin;

```

Début

Ecrire("Entrer un texte ");

Lire(texte);

répéter

Ecrire("1 -pour calculer le nombre de mot palindrome dans un texte");

Ecrire("2 -pour calculer le nombre d'occurrence d'un mot dans un texte");

Ecrire("3 -pour optimiser le nombre de blancs dans un texte");

Ecrire("Choisi une opération");

Lire(choix);

Selon *choix* faire**cas 1 faire**

└ nb ← mot_palindrome(texte);

└ Ecrire("Dans ce texte il existe ",nb,"mot(s) palindrome");

cas 2 faire

└ Ecrire("Entrer un mot ");

└ Lire(mot);

└ nb ← occurrence_mot(texte,mot);

└ Ecrire("Dans ce texte il existe ",nb,"occurrence");

cas 3 faire

└ optimise_blancs(texte);

sinon faire

└ Ecrire("veuillez introduire une nouvelle valeur entre 1 et 3");

jusqu'à *choix* = 0;

Fin.
