Checkpoint Recursion

```
program Palindrome;


function estPalindromeMot(mot: string): boolean;
begin
  // Condition d'arrêt : un mot vide ou un mot contenant un seul caractère est un palindrome
  if (Length(mot) <= 1) then
    estPalindromeMot := true
  else
  begin
    // Comparaison des caractères aux extrémités du mot
    if (mot[1] = mot[Length(mot)]) then
      // Récursivement tester le reste du mot
      estPalindromeMot := estPalindromeMot(Copy(mot, 2, Length(mot) - 2))
    else
      estPalindromeMot := false;
  end;
end;

// Exemple d'utilisation
var
  mot: string;
begin
  // Entrée du mot à vérifier
  write('Entrez un mot : ');
  readln(mot);

  // Vérification si le mot est un palindrome
  if estPalindromeMot(mot) then
```

```
    writeln('Le mot "', mot, '" est un palindrome.')
  else
    writeln('Le mot "', mot, '" n''est pas un palindrome.');
end.
```