La récursivité est un concept qui généralise la notion de suite définie par récurrence.
Une procédure (ou fonction au sens informatique) est *récursive* lorsqu'elle s'appelle elle même.
Certains langages (dont Maple) autorisent cette manière de mettre en oeuvre un algorithme. Il est particulièrement important de prouver que l'algorithme va s'arrêter. En général on introduit pour cela un nombre entier qui doit strictement décroitre d'un appel à l'autre. Un exemple simple est donné par la définition (du cours) des coefficients du binôme.

```
cbin:=proc(n,p)
  if p=0 or p=n then
    1
  else
    cbin(n-1,p-1) + cbin(n-1,p);
  fi;
end:

cbin(5,2);
```

Dans cet exemple le nombre qui diminue strictement à chaque appel est n+p. On peut former une procédure qui rend mieux compte de ce qu'elle fait.

```
nbappel:=0;
cbinbavard:=proc(n,p)
  global nbappel;
  local texte;
  nbappel:=nbappel+1;
  texte:= "appel numero" || nbappel || " , n=" || n || " , p=" || p;
  print(texte);
  if p=0 or p=n then
    1
  else
    cbinbavard(n-1,p-1) + cbinbavard(n-1,p);
  fi;
end:
cbinbavard(3,2);
```

Remarquer qu'il est indispensable d'utiliser une variable globale pour compter le nombre d'appels de la procedure. Remarquer l'opérateur de concaténation || qui permet d'envoyer vers l'écran un joli texte.

Bien noter qu'une procédure récursive doit contenir dès le début une structure de contrôle `if` comprenant une branche qui se termine sans s'appeler. Cette branche correspond à l'initialisation de la récursivité. Sans cela la boucle infinie est assurée. Enregistrez avant d'exécuter.