

I. La récursivité

1. Principe et types de récursivité

- **Récursivité** : Une entité est récursive lorsqu'on l'utilise pour la définir. Une fonction récursive s'« auto-appelle ».
 - **Récursivité terminale** : l'appel récursif est la dernière instruction et est isolé.
 - **Récursivité non terminale** : l'appel récursif n'est pas la dernière instruction ou est dans une expression.

2. Méthode d'écriture d'un programme récursif

- Identifier le ou les cas particulier
- Identifier le cas général qui effectue la récursion

Lors de l'appel récursif, on est utilisateur donc on considère que le programme appelé fonctionne.

3. Conversion récursif terminal en itératif

a. $P \Leftrightarrow Q$

<pre>procédure P(x) debut si C_x alors I sinon J P(f) fin si fin</pre>	<pre>procédure Q(x) debut tant que non C_x faire J x ← f fintantque I fin</pre>
--	---

b. Démonstration par récurrence

Pour 0 appels récursifs, si x est tel que C_x est faux, $P \Leftrightarrow Q$ (on exécute I)

Supposons que $P(x) \Leftrightarrow Q(x)$ pour n appels récursifs. Soit y tel que $P(y)$ fait $n + 1$ appels récursifs.

On a donc $P(y)$:

```
procédure P(y)
debut
  si Cy alors
    Iy
  sinon
    Jy
    P(f)
  fin si
fin
```

$P(f)$ fait n appels donc par hypothèse de rec. donc $P(f) \Leftrightarrow Q(f)$, on peut donc remplacer.

On a également :

```
procédure P(y)
debut
  si Cy alors
    Iy // tjrs exécuté donc on peut enlever le si
  sinon
    // y plus utilisé à partir d'ici, on peut remplacer x par y
    Jy // intégrable dans le tant que
    x ← f // intégrable dans le tant que
    tant que non Cx faire
      Jx
      x ← f
    fintantque
    Ix // tjrs exécuté donc on peut enlever le si
  fin si
fin
```

Finalement, on retrouve $P(y) \Leftrightarrow Q(y)$:

```
procédure P(y)
debut
  tant que non Cy faire
    Jy
    y ← f
  fintantque
  Iy
fin
```