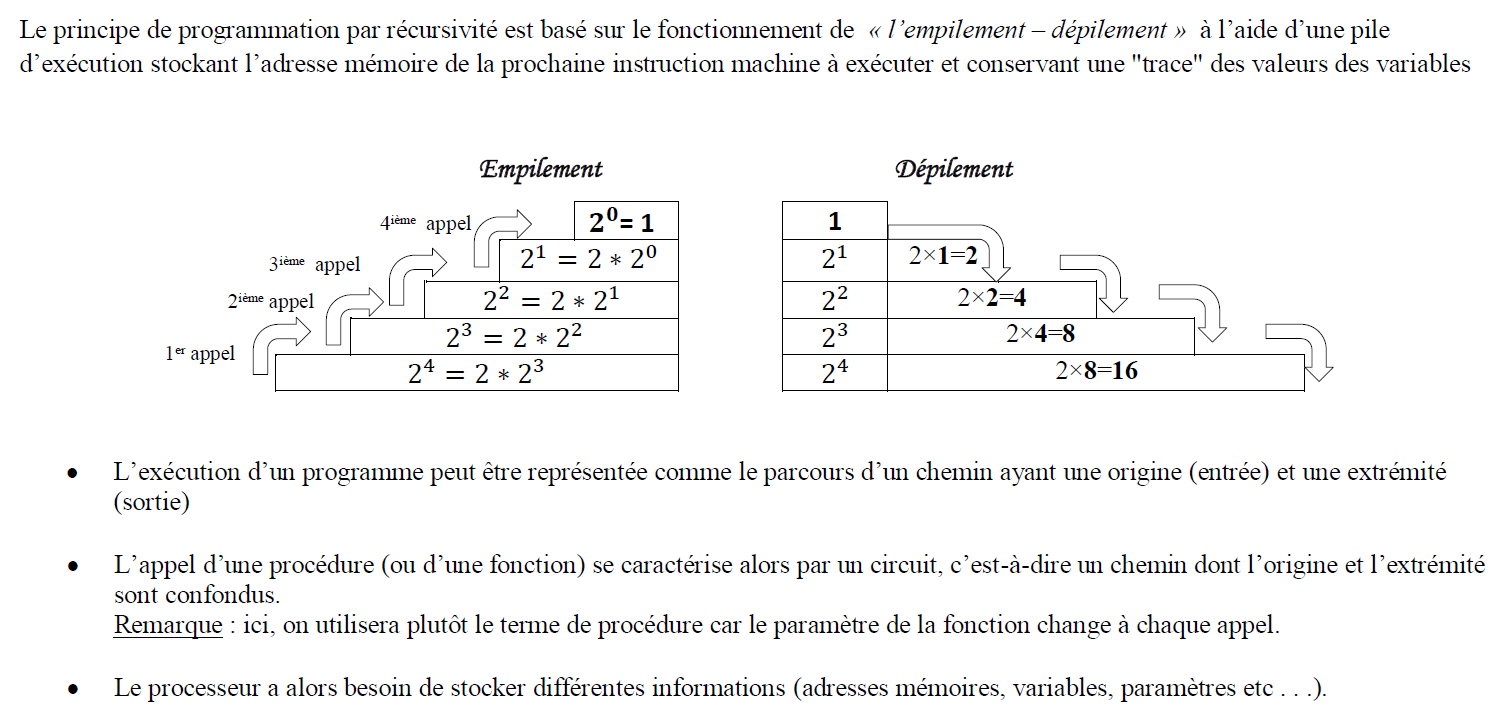
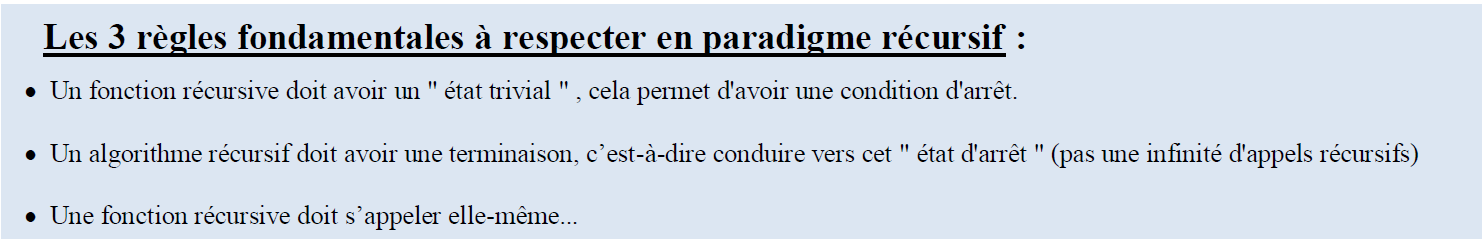


Le principe de programmation par récursivité est basé sur le fonctionnement de *« l’empilement – dépilement »* à l’aide d’une **pile d’exécution** stockant l’adresse mémoire de la prochaine instruction machine à exécuter et conservant une "trace" des valeurs des variables



Pour réaliser tout cela, le processeur gère une ou plusieurs piles dans lesquelles il stocke les adresses de retour des procédures et les valeurs des différentes variables. La récursivité est donc généralement plus lente en raison des frais généraux liés à la maintenance de la pile.



Il faut savoir que ces deux paradigmes de programmation sont équivalents, c’est-à-dire que tout algorithme itératif possède une version récursive, et réciproquement. Le type de langage de programmation utilisé peut conduire à programmer d’une manière ou d’une autre : par exemple Caml est un langage conçu pour exploiter la récursivité. À l'inverse, Python, même s'il l'autorise, ne favorise pas l'écriture récursive (limitation basse (1000 par défaut) du nombre d'appels récursifs).

*Enfin, le choix d'écrire une fonction récursive ou itérative peut dépendre du problème à résoudre : certains problèmes se résolvent particulièrement simplement sous forme récursive)*

La version récursive d’un algorithme est souvent plus succincte et plus lisible

