

**L'opérateur de séquentialité** : il est parfois nécessaire d'effectuer des actions les unes à la suite des autres dans un ordre fixé. L'opérateur `begin` dont la syntaxe est : `(begin <expr1><expr2>... <exprn>` évalue les expressions données en paramètres les unes à la suite des autres et renvoie comme valeur, la valeur de la dernière expressions.

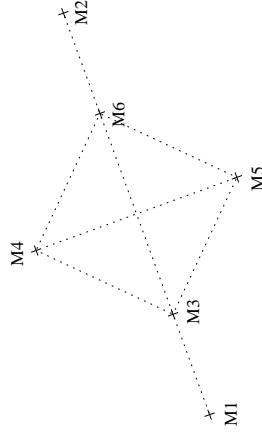
Cet opérateur est utilisé lorsque ce ne sont pas les valeurs des expressions qui importent mais les actions qui sont effectuées. On parle d'effet de bord. C'est le cas avec les fonctions graphiques (ainsi qu'avec les fonctions d'entrée et de sortie) dont les valeurs sont indéfinies mais dont les effets de bords importent.

**Exercice 1 :**

- a) Définir une fonction **carre** qui prend en argument un entier `c` et affiche sur la fenêtre graphique un carré centré au point de coordonnées  $(0, 0)$ , dont le côté et de longueur `c`.
- b) Définir une fonction **anneau**, qui affiche sur la fenêtre graphique un carré centré en  $(0, 0)$ , de côté 400, et qui diminue progressivement jusqu'à se confondre avec l'origine.

**Exercice 2 :** Une fractale.

À partir de deux points  $M_1(x_1, y_1)$  et  $M_2(x_2, y_2)$  du plan, et d'un réel  $k$ , on définit les quatre points  $M_3(x_3, y_3)$ ,  $M_4(x_4, y_4)$ ,  $M_5(x_5, y_5)$  et  $M_6(x_6, y_6)$  tels que :



$$\begin{aligned}
 x_3 &= (3x_1 + x_2)/4 & y_3 &= (3y_1 + y_2)/4 \\
 x_4 &= [2(x_1 + x_2) - k(y_2 - y_1)]/4 & y_4 &= [2(y_1 + y_2) + k(x_2 - x_1)]/4 \\
 x_5 &= [2(x_1 + x_2) + k(y_2 - y_1)]/4 & y_5 &= [2(y_1 + y_2) - k(x_2 - x_1)]/4 \\
 x_6 &= (x_1 + 3x_2)/4 & y_6 &= (y_1 + 3y_2)/4
 \end{aligned}$$

La fractale à l'ordre  $n$ , à partir des points  $M_1$  et  $M_2$  se construit de la manière suivante :

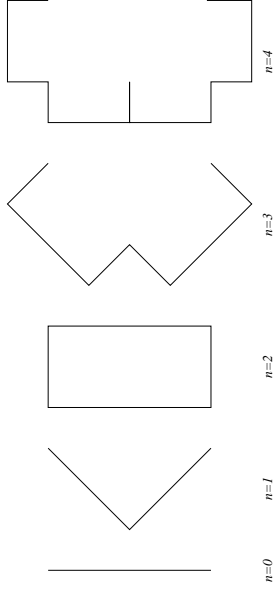
- si  $n = 0$ , c'est le segment  $[M_1, M_2]$  sinon, c'est :
  - la fractale à l'ordre  $n - 1$ , à partir des point  $M_1$  et  $M_2$  et du réel  $k$ ,
  - la fractale à l'ordre  $n - 1$ , à partir des point  $M_3$  et  $M_4$  et du réel  $k$ ,
  - la fractale à l'ordre  $n - 1$ , à partir des point  $M_4$  et  $M_5$  et du réel  $k$ ,
  - la fractale à l'ordre  $n - 1$ , à partir des point  $M_5$  et  $M_6$  et du réel  $k$  et
  - la fractale à l'ordre  $n - 1$ , à partir des point  $M_6$  et  $M_2$  et du réel  $k$ .

Définir une fonction **fractale**, de paramètres  $x_1, y_1, x_2, y_2, k$  et  $n$ . Essayer par exemple (**fractale -200 0 200 0 1 3**).

Remarque : si on oublie le point  $M_5$ , on obtient le flocon de Von Koch.

**Exercice 3 :** Un autre dessin fractal.

Ecrire une fonction **dessin**, de paramètres  $x_1, y_1, x_2, y_2$  et  $n$ , à même de dessiner les figures ci-dessous.



On rappelle que les coordonnées de l'angle droit du triangle rectangle isocèle construit dans le sens direct  $A_1(a_1, b_1)$  et  $A_2(a_2, b_2)$  sont  $(a_1 + b_1 + a_2, (a_2 + b_2 - b_1)/2)$  et  $(a_2 + b_2 + b_1 - a_1)/2$ .

**Fonctions graphiques simplifiées**

Les fonctions qui suivent sont définies à partir des primitives graphiques de DiScheme. Pour les utiliser :  
 - copier le fichier `pack-graph.sc` dans le dossier où se trouve votre programme,

- puis charger ce fichier dans l'environnement global de l'interprète avec la commande `(load "pack-graph`

`(init-graphics)` : ouvre une fenêtre graphique de 600 x 600 points, place l'origine du repère au centre de fenêtre, et initialise la couleur du crayon en noir.

`(init-graphics hauteur largeur)` : idem mais avec une fenêtre graphique de *hauteur* sur *largeur* points. (Maximum x 600).

`(clear-graphics)` : efface le contenu de la fenêtre graphique.

`(black-pen)` : sélectionne le crayon noir.

`(white-pen)` : sélectionne le crayon blanc.

`(red-pen)` : sélectionne le crayon rouge.

`(green-pen)` : sélectionne le crayon vert.

`(blue-pen)` : sélectionne le crayon bleu.

`(yellow-pen)` : sélectionne le crayon jaune.

`(orange-pen)` : sélectionne le crayon orange.

`(purple-pen)` : sélectionne le crayon violet.

`(set-rgb-color r g b)` : affecte au crayon la couleur déterminée par les trois valeurs  $r, g$  et  $b$ . Ces trois valeurs sont réels de l'intervalle  $[0, 1]$  est représentent les quantités de rouge, de vert et de bleu de la couleur.

`(position-pen x y)` : place le crayon au point de coordonnées  $(x, y)$ .

`(draw-point x y)` : dessine le point de coordonnées  $(x, y)$  dans la couleur courante du crayon.

`(clear-point x y)` : efface le point de coordonnées  $(x, y)$ .

`(draw-line-to x y)` : trace une ligne depuis la position courante du crayon jusqu'au point de coordonnées  $(x, y)$  qui devient la nouvelle position du crayon.