Mouvement de Tortue

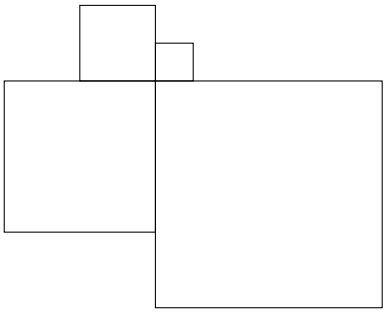
On s'intéresse ici au module `Turtle`.
On commencera donc par importer ce module via l'instruction :

```
1 from turtle import *
```

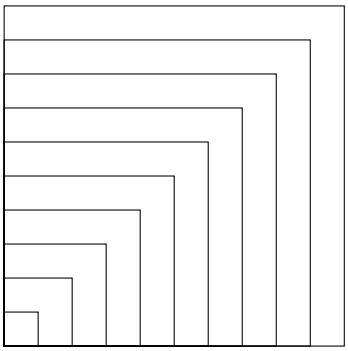
1. Recopier le programme suivant et observer ce qu'il affiche.

```
1 reset()
2 forward(200)
3 left(120)
4 forward(200)
5 left(120)
6 forward(200)
```

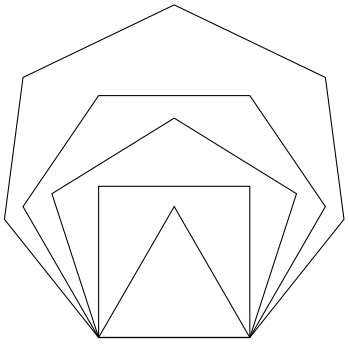
2. Remettre à zéro le contenu de la fenêtre graphique, puis faire dessiner par la tortue un carré.
3. Définir une fonction `carre` qui prend en argument un entier n et qui renvoie un carré de taille n.
4. A l'aide de la fonction `carre`, rédiger un script pour reproduire la figure suivante :



5. A l'aide de la fonction `carre` et d'une boucle `for`, rédiger un script pour reproduire la figure suivante :



6. Augmenter la vitesse de la tortue jusqu'à sa vitesse maximale puis, à l'aide d'une boucle `for`, faire avancer la tortue de 1, puis de 2, 3, 4, ..., 500 pas en tournant d'un angle de 91 degrés entre chaque déplacement.
7. Définir une fonction `polygone(n,l)` qui trace un polygone régulier à n côtés, chacun de longueur l, puis reproduire à l'aide de cette fonction la figure suivante :



Le triangle de Sierpiński est une fractale qui s'obtient à partir d'un triangle plein par une infinité de répétitions consistant à diviser par 2 la taille du triangle puis à les accoler en trois exemplaires par leur sommets pour former un nouveau triangle. Ci-dessous un exemple des trois premières générations de cette transformation :

Pour colorer l'intérieur d'une courbe fermée, il faut précéder le début d'un tracé par la commande `begin_fill()` et terminer celui-ci par `end_fill()`. Par exemple :

```
1 begin_fill()
2 circle(100)
3 end_fill()
```

8. (a) Définir une fonction `sierp1(l)` qui dessine la première étape de la construction du triangle de Sierpiński, c'est-à-dire un triangle plein de longueur ℓ.
   (b) A l'aide de la fonction précédente, définir une fonction `sierp2(l)` qui dessine la deuxième étape de la construction du triangle de Sierpiński.
   (c) A l'aide de la fonction précédente, définir une fonction `sierp3(l)` qui dessine la troisième étape de la construction du triangle de Sierpiński.
9. Définir la fonction `sierpinski(n,l)` qui dessine la génération d'ordre n du triangle de Sierpiński avec une longueur de côté égale à ℓ, puis utiliser cette fonction pour effectuer le tracé avec $n = 5$.

Annexe : Diverses commandes du module turtle

Déplacement de la tortue

- `forward(n)` avance la tortue de n pas dans la direction qui lui fait face ;
- `backward(n)` recule la tortue de n pas ;
- `left(n)` tourne la tortue d'un angle de n degré dans le sens trigonométrique ;
- `right(n)` tourne la tortue d'un angle de n degré dans le sens horaire ;
- `speed(n)` modifie la vitesse de déplacement de la tortue de $n = 1$ (lent) à $n = 10$ (rapide) ; $n = 0$ (le plus rapide) supprime toute animation. Par défaut $n = 3$.

Tracé du chemin de la tortue

- `penup()` soulève le croyan de sorte que le déplacement de la tortue ne trace pas de trait ;
- `pendown()` baisse le croyan de sorte que le déplacement de la tortue trace un trait ;
- `pensize(n)` modifie l'épaisseur de la ligne que la tortue trace en se déplaçant (par défaut $n = 1$) ;
- `pencolor(c)` modifie la couleur de la ligne que la tortue trace ;

Commandes diverses

- `position()` retourne les coordonnées (x, y) de la tortue ;
- `heading()` retourne l'angle de la tortue par rapport à l'horizontale ;
- `setposition(x,y)` déplace la tortue au point de coordonnées (x, y) ;
- `distance(x,y)` retourne la distance entre la tortue et le point de coordonnées (x, y).
- `towards(x,y)` retourne l'angle avec l'horizontale que fait la droite reliant la tortue au point de coordonnées (x, y) ;
- `home()` déplace la tortue au point de coordonnées $(0, 0)$ en la dirigeant vers l'est ;
- `reset()` efface tous les tracés et replace la tortue à sa position initiale.