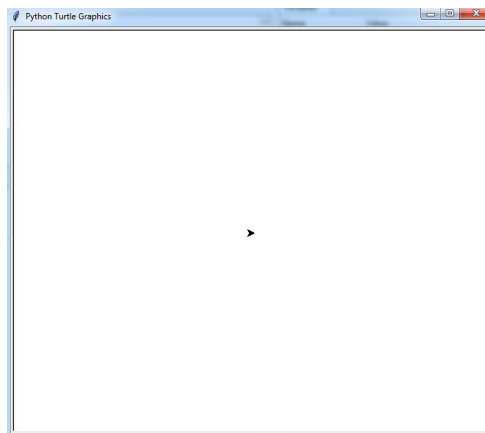


TP turtle

La tortue désigne un robot virtuel capable de se déplacer dans un plan en laissant une trace ou non de son passage. Elle a été inventée dans les années 1960 au MIT par Seymour Papert qui a conçu un langage informatique [LOGO](http://fr.wikipedia.org/wiki/Logo_%28langage%29) (http://fr.wikipedia.org/wiki/Logo_%28langage%29) destiné aux enfants. De nombreux langages informatiques ont repris cette idée de tortue et Python n'échappe pas à la règle.

- Le **module** `turtle` propose un environnement graphique et un ensemble de **fonctions** permettant le déplacement de la tortue. En se déplaçant, la tortue laisse une trace derrière elle comme un escargot.
- La tortue apparaît à l'écran sous la forme d'un triangle (voir dessin ci-dessous). Elle est repérée par des coordonnées (x, y) mais aussi par un angle en degrés entre la direction vers laquelle elle "regarde" et l'axe horizontal.



Pendant ce tp, vous allez écrire des fonctions qui dessinent des formes géométriques, en déplaçant la tortue. Une bonne connaissance du module `turtle`, et des **boucles**, permet de réaliser des [formes très abouties](https://www.google.com/search?q=turtle+python&tbm=isch&ved=2ahUKEwj263JhdpAhWN8IUkHYwkBTAQ2-cCegQIABAA&oq=turtle+python&gs_lcp=CgNpbWcQAzIECCMQJzICCAAYBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4ysclient=img&ei=NoLRXvPoO43hlwSMYzSAAw&bih=626&biw=1366&client=firefox-b-d)

([https://www.google.com/search?q=turtle+python&tbm=isch&ved=2ahUKEwj263JhdpAhWN8IUkHYwkBTAQ2-cCegQIABAA&oq=turtle+python&](https://www.google.com/search?q=turtle+python&tbm=isch&ved=2ahUKEwj263JhdpAhWN8IUkHYwkBTAQ2-cCegQIABAA&oq=turtle+python&gs_lcp=CgNpbWcQAzIECCMQJzICCAAYBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4ysclient=img&ei=NoLRXvPoO43hlwSMYzSAAw&bih=626&biw=1366&client=firefox-b-d)

[gs_lcp=CgNpbWcQAzIECCMQJzICCAAYBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4ysclient=img&ei=NoLRXvPoO43hlwSMYzSAAw&bih=626&biw=1366&client=firefox-b-d](https://www.google.com/search?q=turtle+python&tbm=isch&ved=2ahUKEwj263JhdpAhWN8IUkHYwkBTAQ2-cCegQIABAA&oq=turtle+python&gs_lcp=CgNpbWcQAzIECCMQJzICCAAYBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4yBAgAEB4ysclient=img&ei=NoLRXvPoO43hlwSMYzSAAw&bih=626&biw=1366&client=firefox-b-d)), notamment ayant des

structures [fractales](https://fr.wikipedia.org/wiki/Fractale) (<https://fr.wikipedia.org/wiki/Fractale>), comme [ici](https://www.google.com/search?q=fractal+python+turtle&tbm=isch&ved=2ahUKEwjWtdvChtrpAhUFw4UKHeuVDvEQ2-cCegQIABAA&oq=turtle+python+frac&gs_lcp=CgNpbWcQARgAMgYIABAIEB46BAgjECc6AggAOgQIABAEogQIABAYUL2EDVjSiw1g5JkNaABwAHgAgAFci/sclient=img&ei=NYPRXpbHHYWGlwTrq7qIDw&bih=626&biw=1366&client=firefox-b-d) ([https://www.google.com/search?q=fractal+python+turtle&tbm=isch&ved=2ahUKEwjWtdvChtrpAhUFw4UKHeuVDvEQ2-cCegQIABAA&oq=turtle+python+frac&](https://www.google.com/search?q=fractal+python+turtle&tbm=isch&ved=2ahUKEwjWtdvChtrpAhUFw4UKHeuVDvEQ2-cCegQIABAA&oq=turtle+python+frac&gs_lcp=CgNpbWcQARgAMgYIABAIEB46BAgjECc6AggAOgQIABAEogQIABAYUL2EDVjSiw1g5JkNaABwAHgAgAFci/sclient=img&ei=NYPRXpbHHYWGlwTrq7qIDw&bih=626&biw=1366&client=firefox-b-d)

[gs_lcp=CgNpbWcQARgAMgYIABAIEB46BAgjECc6AggAOgQIABAEogQIABAYUL2EDVjSiw1g5JkNaABwAHgAgAFci/sclient=img&ei=NYPRXpbHHYWGlwTrq7qIDw&bih=626&biw=1366&client=firefox-b-d](https://www.google.com/search?q=fractal+python+turtle&tbm=isch&ved=2ahUKEwjWtdvChtrpAhUFw4UKHeuVDvEQ2-cCegQIABAA&oq=turtle+python+frac&gs_lcp=CgNpbWcQARgAMgYIABAIEB46BAgjECc6AggAOgQIABAEogQIABAYUL2EDVjSiw1g5JkNaABwAHgAgAFci/sclient=img&ei=NYPRXpbHHYWGlwTrq7qIDw&bih=626&biw=1366&client=firefox-b-d)), notamment ayant des

Découverte du module `turtle`

Pour utiliser les fonctions du module, il faut l'importer grâce à la directive `import` (voir cours sur les bibliothèques - bloc6)

Le module `turtle` possède de nombreuses fonctions dont les principales sont :

- `clear` qui efface la zone de dessin
- Paramétrages du crayon :
 - `down` qui abaisse le stylo
 - `up` qui relève le stylo
 - `pensize` qui change l'épaisseur du trait

- `pencolor` qui change la couleur ("red", "green", "blue"... ou un triplet de paramètres (r, g, b))

- Déplacement de la tortue:
 - `forward` qui avance d'un nombre de pas donné
 - `backward` qui recule
 - `right` qui tourne vers la droite d'un angle donné (en degrés)
 - `left` qui tourne vers la gauche.
 - `goto` qui déplace la tortue jusqu'à un point de coordonnées (x, y)

A l'aide des fonctions `help` et `dir`, prenez une dizaine de minutes pour prendre connaissance du module `turtle` et faire quelques essais des fonctions ci-dessus.

Travail demandé

Remarque importante : Pour écrire une fonction, il est vivement recommandé, lorsque c'est possible, d'utiliser une fonction écrite dans une question précédente

1 - Dessiner un carré

Pour tracer un carré au sol, un homme ferait les étapes suivantes :

- * Avancer de 100 pas,
- * Tourner sur la gauche d'un angle droit,
- * Avancer de 100 pas,
- * Tourner sur la gauche d'un angle droit,
- * Avancer de 100 pas,
- * Tourner sur la gauche d'un angle droit,
- * Avancer de 100 pas,
- * Tourner sur la gauche d'un angle droit.

En théorie, on doit se retrouver au point de départ (position et orientation)

Traduire cet algorithme pour la tortue en python et tester

2 - Dessiner un carré - Bis

Nous pouvons constater qu'il devient fastidieux de réécrire plusieurs fois l'instruction "Avancer de 100 pas" et "Tourner sur la gauche d'un angle droit" pour écrire le programme de construction d'un carré. Heureusement il est possible de n'écrire qu'une fois les instructions que l'on doit répéter et de dire combien de fois on veut que ces instructions soient répétées.

Simplifier le programme de la construction d'un carré, en utilisant une boucle.

3 - Dessiner un carré - Ter

Encapsuler votre code précédent dans une fonction `carré` avant la spécification suivante :

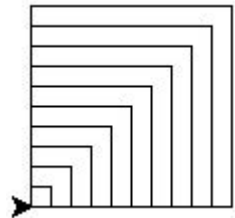
```
In [ ]: def carré(pas):  
        """  
        Description de la fonction : Déplace la tortue de sorte qu'elle dessine u  
n carré  
et revienne au point de départ (même positio  
n et orientation)  
pas(int) : côté du carré  
return : None  
        """  
        # A compléter
```

4 - Dessiner des carrés emboîtés

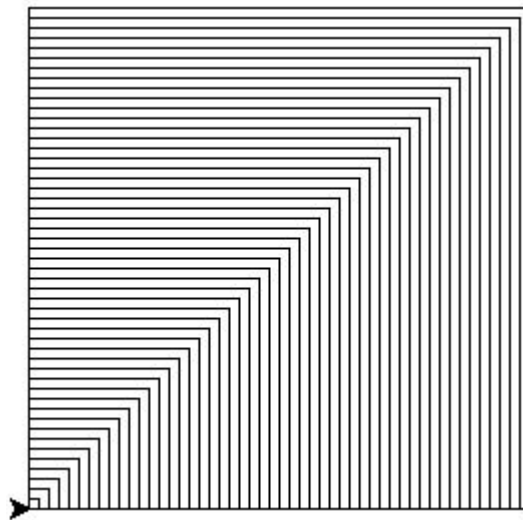
Ecrire le code de la fonction ci-dessous. Des exemples d'appels de cette fonction sont donnés juste après.

```
In [ ]: def carreEmboite(cote,nb):  
        """  
        Description de la fonction : Déplace la tortue de sorte qu'elle dessine u  
ne succession  
                                de carrés emboîtés régulièrement espacés  
pas(int) : côté du plus petit carré  
nb(int): nombre total de carrés  
return : None  
        """  
        #A compléter
```

carreEmboite(10,10)



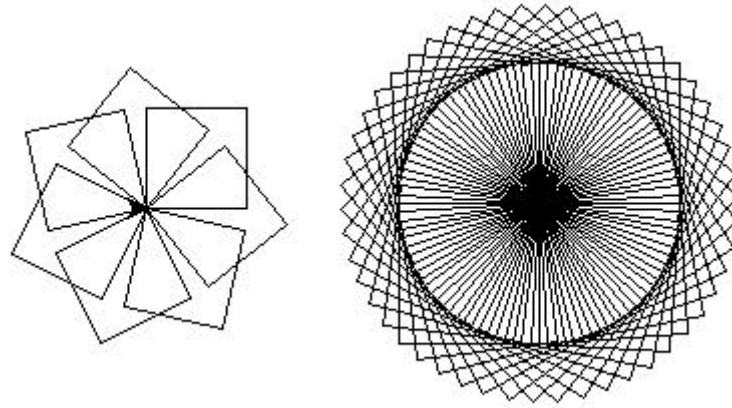
carreEmboite(5,50)



5 - Dessiner des carrés tournants

Ecrire le code de la fonction ci-dessous. Des exemples d'appels de cette fonction sont donnés juste après.

```
In [ ]: def carreTournant(cote,nb):  
        """  
        Description de la fonction : Déplace la tortue de sorte qu'elle dessine u  
ne succession  
                                de carrés tournants régulièrement espacés  
pas(int) : côté du plus petit carré  
nb(int): nombre total de carrés  
return : None  
        """  
        #A compléter
```



6 - Dessiner une ligne de carrés

Ecrire le code de la fonction ci-dessous. Des exemples d'appels de cette fonction sont donnés juste après.

```
In [ ]: def ligne2carre(nb,cote,espacement):
    """
    Description de la fonction : Déplace la tortue de sorte qu'elle dessine u
    ne ligne de carré
    nb(int) : nombre de carrés dessinés
    cote(int): côté du carré
    espacement(int) : espacement entre 2 carrés successifs
    return : None
    """
    #A compléter
```

ligne2carre(10,15,2)

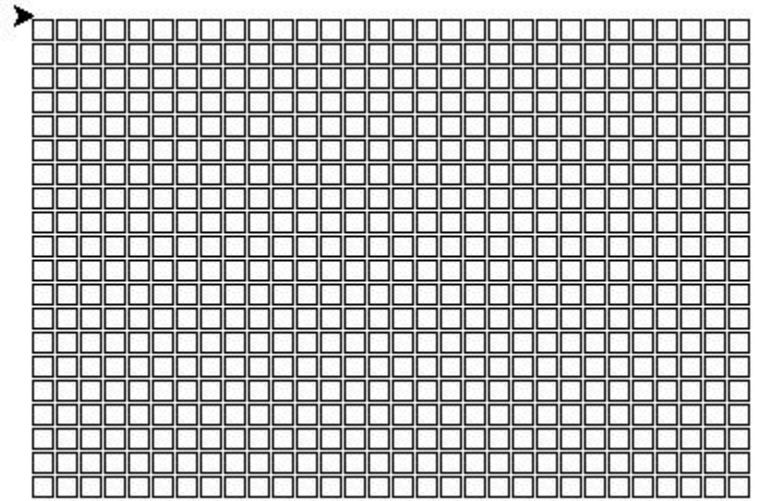
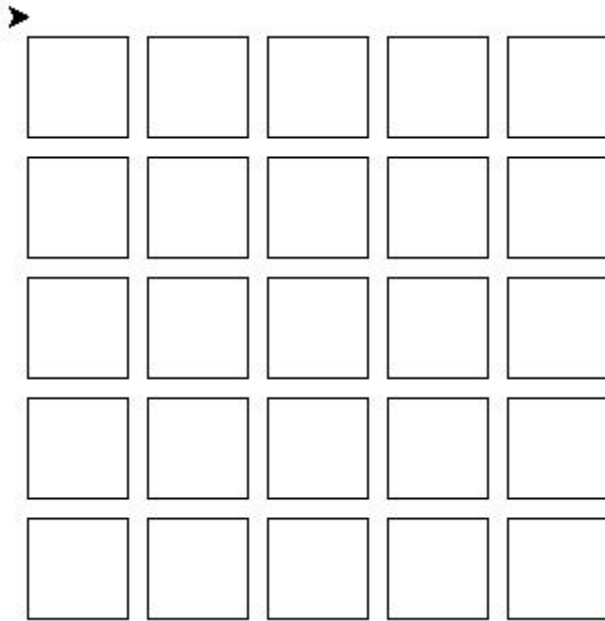
ligne2carre(20,6,10)



7 - Dessiner un damier

Ecrire le code de la fonction ci-dessous. Des exemples d'appels de cette fonction sont donnés juste après.

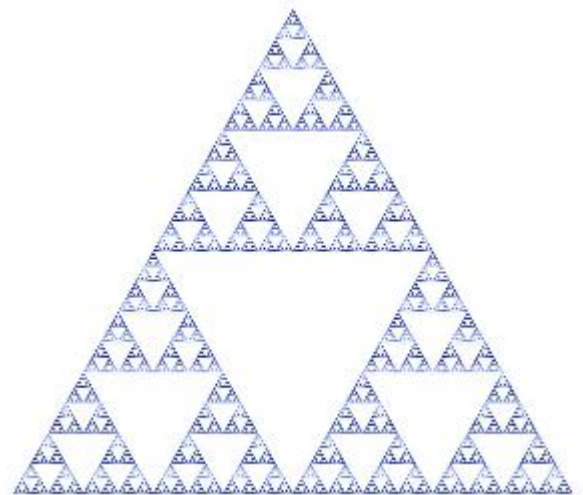
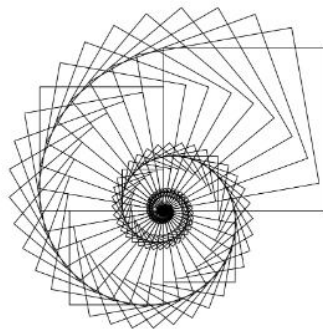
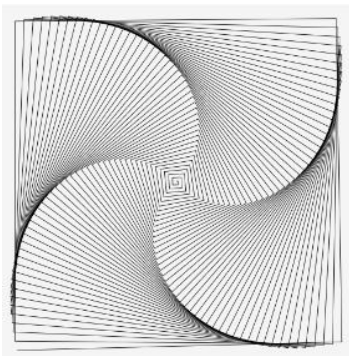
```
In [ ]: def damier(longueur,hauteur,cote,espacement):
    """
    Description de la fonction : Déplace la tortue de sorte qu'elle dessine u
    n damier
    longueur(int) : nombre de carrés par ligne
    hauteur(int) : nombre de carrés par colonne
    cote(int): côté du carré
    espacement(int) : espacement entre 2 carrés successifs
    return : None
    """
    #A compléter
```



Pour aller plus loin...

Vous pouvez programmer la tortue pour qu'elle réalise les belles fractales ci-dessous...

[Triangle de Sierpiński \(https://fr.wikipedia.org/wiki/Triangle_de_Sierpi%C5%84ski\)](https://fr.wikipedia.org/wiki/Triangle_de_Sierpi%C5%84ski)



Quelques fractales animées...

