

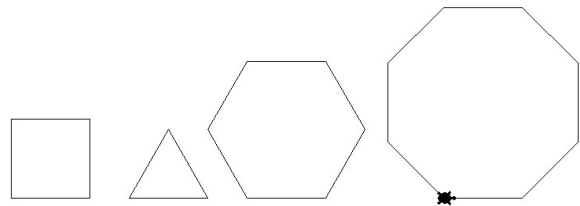
TP PYTHON TURTLE

Sauf indication contraire, vous ne devez pas utiliser la fonction **circle(rayon[,angle],pas)** pour tracer un polygone.

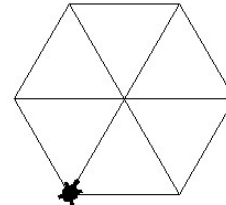
I. Création de figures de base.

Créer un programme pour tracer :

- 1°) Un carré.
- 2°) Un triangle équilatéral.
- 3°) Un hexagone régulier.
- 4°) Un octogone régulier.
- 5°) Les quatre figures l'une à côté de l'autre.



II. Créer un programme pour tracer un hexagone régulier et ses diagonales de centre le centre de l'hexagone.

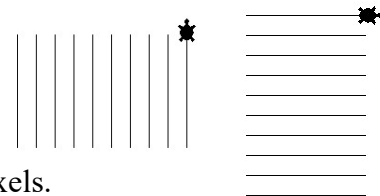


III. Quadrillages de 10 traits.

Créer un programme pour tracer :

- 1°) Un quadrillage vertical.
- 2°) Un quadrillage horizontal.

Chaque trait à une longueur de 150 pixels et est espacé de 20 pixels.

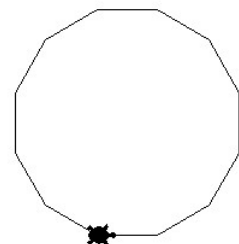


IV. Polygone régulier à n côtés.

Créer un programme pour tracer un polygone régulier à n côtés. Le programme devra demander la longueur d'un côté et le nombre de côtés du polygone.

Exemple :

```
>>>
Entrer la longueur d'un côté : 50
Entrer le nombre de côtés : 12
>>>
```



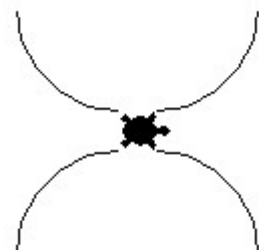
V. Cercle : sens direct ou indirect ; pôle nord ou sud.

Créer un programme pour tracer la figure ci-contre.

Chaque départ de demi-cercle correspond à un sommet d'un carré de côté 20 centré en (0;0).

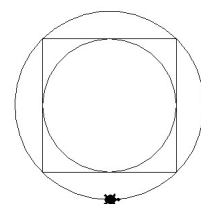
Aide :

- l'instruction `circle(r)` permet de tracer un cercle de rayon r à la position courante. La tortue trace le cercle en commençant :
 - dans le sens direct ; départ par le pôle sud si $r > 0$;
 - dans le sens indirect ; départ par le pôle nord si $r < 0$.
- l'instruction `circle(r,a)` permet de tracer un arc cercle, d'angle au sommet a .



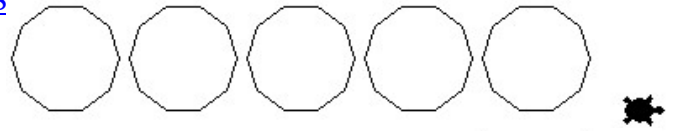
VI. Cercles et carré.

Créer le programme pour tracer la figure ci-contre avec un carré de côté 200 pixels.

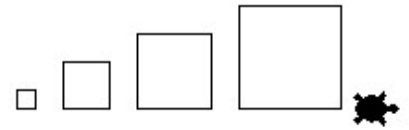


VII. Suites de figures.

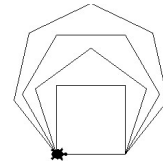
1°) Créer le programme pour tracer une suite de cinq décagones réguliers de côté 20 pixels dont les bases sont espacées de 50 pixels.



2°) Créer le programme pour tracer une suite de quatre carrés dont les côtés augmentent de 15 pixels au fur et à mesure. Le premier carré a un côté de 10 pixels. On pourra créer la fonction `carre(c)` qui trace un carré de côté `c` à la position courante.

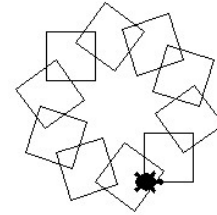


3°) Créer le programme pour tracer une suite de quatre polygones réguliers de côtés 100 pixels mais dont le nombre de côté augmente au fur et à mesure.

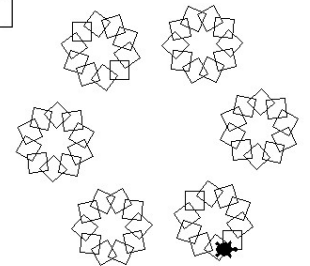


Écrire le programme le plus court possible en utilisant deux boucles imbriquées.

4°) a) Créer le programme pour tracer une suite de dix carrés de côté 40 pixels. On pourra créer la fonction `carre(c)` qui trace un carré de côté `c` à la position courante.



b) En utilisant le programme précédent, construire la mosaïque ci-contre.



VIII. Dessins aléatoires (mouvement Brownien).

1°) Dessiner un cercle de rayon 100 et de centre (0,0)

2°) Se placer sans dessiner de trait au centre du cercle

3°) Écrire un programme qui choisit aléatoirement un angle (entre 0 et 359°) et une longueur (entre 5 et 10), qui tourne la tortue de cet angle ; avance de cette longueur et trace un trait tant que la tortue ne sort pas du cercle. (l'angle et la longueur seront réactualisés à chaque étape).

a) Utiliser la fonction `distance()` du module turtle.

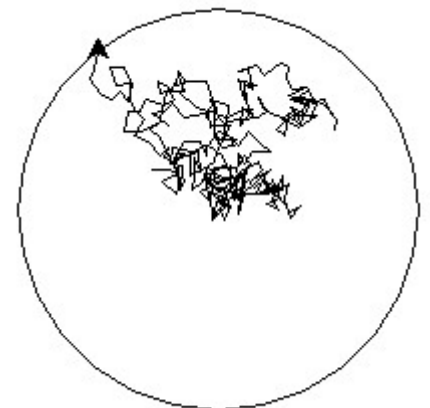
b) Ne pas utiliser la fonction `distance()` mais les fonctions `xcor()` et `ycor()`. On rappelle que dans un repère orthonormé (O, I, J) , la distance entre deux points $A(x_A; y_A)$ et $B(x_B; y_B)$ est

$$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}.$$

c) Ne pas utiliser ni la fonction `distance()`, ni les fonctions `xcor()` et `ycor()`. Vous créez deux variables abscisse et ordonnée ; Vous pourrez utiliser :

➤ les fonctions aléatoires que l'on importe via : **from random import ***

➤ les fonctions sinus et cosinus que l'on importe via : **from math import ***. Pour calculer le cosinus d'un angle en degrés la commande est `cos(radians(ang))`. Même chose avec le sinus.



IX. Quelques (petits) défis.

Créer les programmes pour créer les figures suivantes.

1°) Le yin et le yang.	2°) Une cible	3°) Une spirale	4°) Une rosace
