

Nom, prénom, classe :

Evaluation de python – avec turtle

Voici deux figures : une sans remplissage, et une avec.

Vous devez réaliser ce travail de la manière suivante :

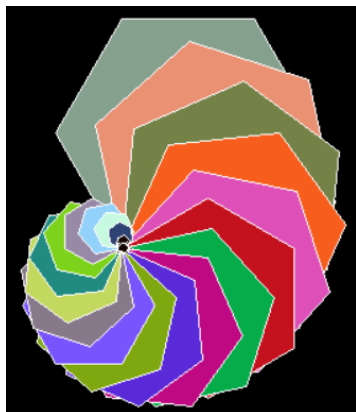
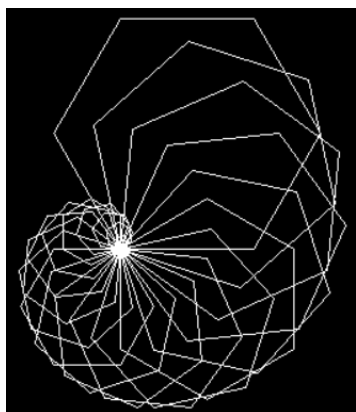
1 (10 pts) – Réaliser la figure sans remplissage. Pour chaque étoile successive, on augmente la taille et on la redessine.

Puis dans l'ordre de votre choix :

2 (3 pts) – Écrivez une **fonction** pour le tracé de chaque polygone (cf exemple du cours ).

3 (4 pts) – Rendre la figure paramétrable avec **trois paramètres**. Votre application doit demander à l'utilisateur deux valeurs permettant de personnaliser le résultat obtenu. Elles seront la **longueur du côté du plus grand polygone**, le **nombre de côtés**, et le **nombre de polygones pour faire un tour**.

4 (3 pts) – Effectuer les **remplissages** au fil du tracé en partant du plus grand polygone.



, et avec 20 côtés :



Exemple de programme à compléter, à télécharger dans votre cours Moodle : (Fichier_depart_evaluation.py)

```
from turtle import *
from random import *

shape("turtle")
screensize(1500,1500) # ne fonctionne pas dans Trinket
reset()
shapexsize(.5,.5,0)
bgcolor("black")
color("white")
clear()
speed(1000)
colormode(255)

# VOTRE PROGRAMME ICI
done()
```

Aides :

- Couleur au hasard, avec au début « colormode(255) »
`color(randint(0, 255),randint(0, 255),randint(0, 255))`
- Remplissage
`fillcolor("white")` # c'est un exemple
`begin_fill()`
votre dessin à remplir
`end_fill()`

Aides (cela signifie que vous ne savez pas encore faire !!!) :

Soient les trois valeurs fournies par l'utilisateur dans trois *input* :

- *cotes* : le nombre de cotés
- *taillem* : la taille du coté du plus grand polygone
- *nbpol* : le nombre de cotés pour faire un tour

Dans le **for in range**, il faut aller de la *taillem* à 0 par incrément de : $-taillem // nbpol$
// est une division euclidienne pour obtenir un entier.

L'angle entre deux polygones est : $360 / nbpol$

La fonction de tracé de polygone est dans l'exemple de figure  du cours :

```
def poly(long,cot):  
    for j in range (cot):  
        forward(long)  
        left(360/cot)
```

Exemple de départ : je prends le programme de figure  du cours et je tente de faire un tour (360 degrés) avec 20 octogones :

```
from turtle import *  
from random import *  
  
shape("turtle")  
screensize(1000,1000) # ne fonctionne pas dans Trinket  
reset()  
shapexsize(.5,.5,0)  
bgcolor("black")  
color("white")  
clear()  
speed(10)  
colormode(255)  
  
def poly(long,cot):  
    for j in range (cot):  
        forward(long)  
        left(360/cot)  
  
for i in range (20):  
    poly(50,8)  
    right(360/20)  
  
done()
```

Puis je diminue l'octogone à chaque tracé :

```
for i in range (20):  
    poly(50*(1-i/20),8)  
    right(360/20)
```

Remarque : $1-i/20$ permet d'aller de 1 pour $i=0$, jusqu'à 0 pour $i = 20$. A l'évaluation, je ne demanderai pas cela.

Ensuite on demande à l'utilisateur les paramètres :

```
cotes = int(input("Combien de cotés ? "))  
taillem = int(input("Quelle taille maximale pour le coté ? "))  
nbpol = int(input("Combien de polygone (pour un tour) ? "))
```

Et on les utilise :

```
for i in range(taillem,0,-taillem//nbpol):  
    poly(i,cotes)  
    end_fill()  
    right(360/nbpol)
```

*Remarque : dans le for in range, on va en décroissant de *taillem* à 0 par pas de $-taillem//nbpol$ car cela divise *taillem* par le nombre de polygones à tracer (par exemple, si on en veut 10, on enlève *taillem*/10 à chaque fois), et on utilise // pour avoir une division entière ou euclidienne.*

Reste à colorier !!!

NE DESCENDEZ PAS VOIR LA SOLUTION !!! SINON VOUS PERDREZ TOUT LE BENEFINE DE L'EXERCICE !!!

Une solution... :

```
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 22 11:29:55 2025

@author: vince
"""
from turtle import *
from random import *

shape("turtle")
screensize(1000,1000) # ne fonctionne pas dans Trinket
reset()
shapeseize(.5,.5,0)
bgcolor("black")
color("white")
clear()
speed(0)
colormode(255)

cotes = int(input("Combien de cotés ? "))
taillem = int(input("Quelle taille maximale pour le coté ? "))
nbpol = int(input("Combien de polygone (pour un tour) ? "))

angle = 360/nbpol

def poly(long,cot):
    for j in range (cot):
        forward(long)
        left(360/cot)

for i in range(taillem,0,-taillem//nbpol):
    fillcolor(randint(0, 255),randint(0, 255),randint(0, 255))
    begin_fill()
    poly(i,cotes)
    end_fill()
    right(angle)

done()
```