

OBJECTIFS : L'objectif principal de ce second TP est d'utiliser la structure « *for .. in range(...)* : » vue en cours dans des situations différentes.

DOCUMENT A RENDRE : Ce travail est évalué. Vous en rédigerez un compte-rendu numérique en utilisant un logiciel de traitement de texte (*Word ou Libre Office*). Le fichier constitué sera appelé *tp2.doc* ou *tp2.odt* et devra être transféré en fin d'activité **par l'intermédiaire** du site *nsibranly.fr* : se loguer et transférer en utilisant le code **tp2** . Ce compte-rendu contiendra :

- les réponses aux différentes questions posées,
- les captures d'écran **des morceaux de codes écrits et celles des résultats des exécutions** données dans le shell. Pour faire ces captures, utiliser *l'Outil Capture d'écran* de Windows.

1. Code donnant des exemples de conversion monétaire entre Yuan chinois et € :

La monnaie chinoise est le Yuan. Le taux de conversion est actuellement : 1 Yuan = 0.12 € .

⇒ Ecrire un code comprenant une boucle *for* et qui permet d'afficher les 10 lignes ci-contre :

```
1 Yuan(s) = 0.12 €
2 Yuan(s) = 0.24 €
3 Yuan(s) = 0.36 €
4 Yuan(s) = 0.48 €
5 Yuan(s) = 0.6 €
6 Yuan(s) = 0.72 €
7 Yuan(s) = 0.84 €
8 Yuan(s) = 0.96 €
9 Yuan(s) = 1.08 €
10 Yuan(s) = 1.2 €
```

2. Code précédent mais amélioré :

⇒ Modifier le code précédent, afin qu'il puisse afficher les 4 lignes ci-contre :

Info : $10^1 = 10^{**}1 = 10$
 $10^2 = 10^{**}2 = 100$
 $10^3 = 10^{**}3 = 1000$

```
10 Yuan(s) = 1.2 €
100 Yuan(s) = 12.0 €
1000 Yuan(s) = 120.0 €
10000 Yuan(s) = 1200.0 €
```

3. Code qui affiche des nombres au carré :

⇒ Ecrire un code qui demande à l'utilisateur combien de nombres au carré il faut afficher. Après saisi de ce nombre, le code les affiche à l'écran. Par exemple :

```
Combien de nombres au carré on affiche ? :10
1  4  9  16  25  36  49  64  81
```

4. Code précédent, mais amélioré :

⇒ Modifier le code précédent, afin qu'il puisse afficher en plus la somme des nombres affichés. Par exemple :

```
Combien de nombres au carré on affiche ? :10
1  4  9  16  25  36  49  64  81
La somme de ces nombres est : 285
```

Info : Pour demander un retour à la ligne dans la fonction print(), on place `\n` dans la chaîne de caractère à afficher. Par exemple :

```
print("\nLa somme de ces nombres est :",somme)
```

5. Code à compléter

⇒ Le code ci-contre est incomplet. Il manque l'argument du 2^{ième} appel à la fonction range().

```
n = int(input("Combien ? : "))
for i in range(1,n+1) :
    for j in range( ) :
        print("*",end="")
    print()
```

```
Combien ? : 5
```

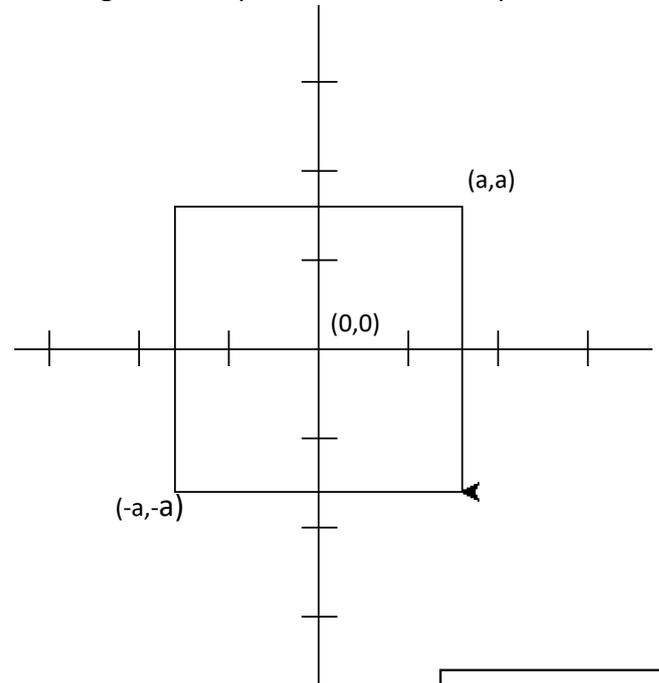
```
*
**
***
****
*****
```

Que faut-il mettre entre parenthèses pour obtenir le motif ci-contre à l'exécution avec la valeur 5 ?

Ecrire le code et essayer

6. Dessin d'un carré par le positionnement de ses côtés :

⇒ Ecrire un code qui demande à l'utilisateur la demi-largeur a en pixel d'un carré et qui le trace ensuite en utilisant la bibliothèque *turtle*. Utilisez pour cela les méthodes `goto(x,y)`, `up()` et `down()` à bon escient.

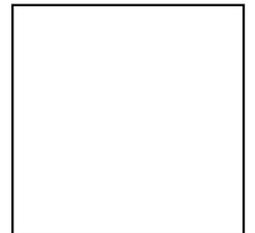


7. Dessin d'un carré par répétition d'une même séquence :

⇒ Ecrire un code qui dessine un carré de côté a également mais en répétant autant de fois que possible la même séquence.

Rappeler :

- quelle est la méthode utilisée pour avancer d'une distance donnée ?
- quelle méthode permet de tourner le crayon d'un angle en degré ?

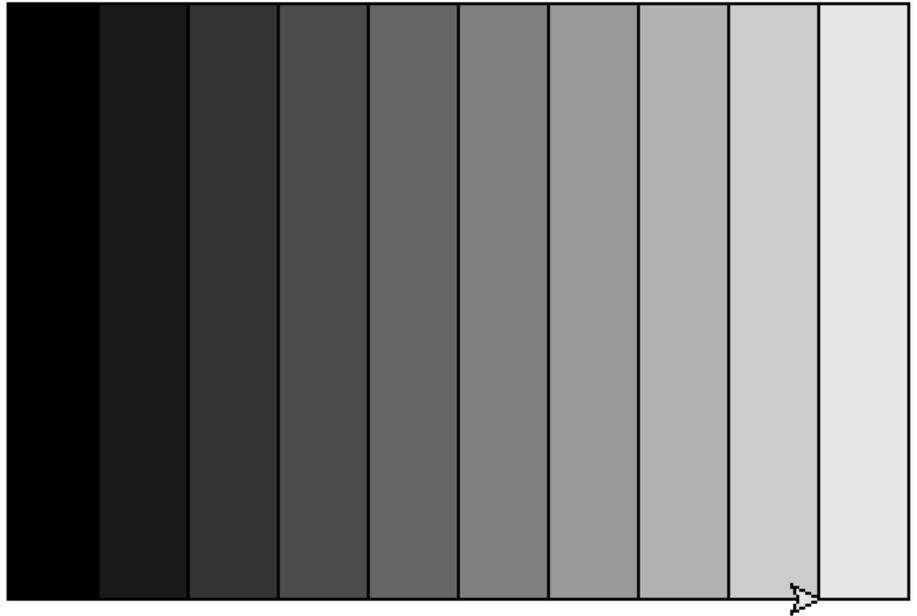


8. Dégradé de rectangles gris :

⇒ Ecrire un code qui permette d'afficher la figure ci-contre :

Le premier rectangle possède une largeur de 30 pixels et une hauteur de 200 pixels et est rempli de la couleur noire.

Chaque rectangle est rempli uniformément d'une couleur d'abord noire puis par un gris plus clair proportionnel à sa position. C'est en fait le même rectangle qui est décalé en abscisse de la largeur du premier. Utiliser une séquence de boucle répétitive pour tracer cette figure.

Infos :

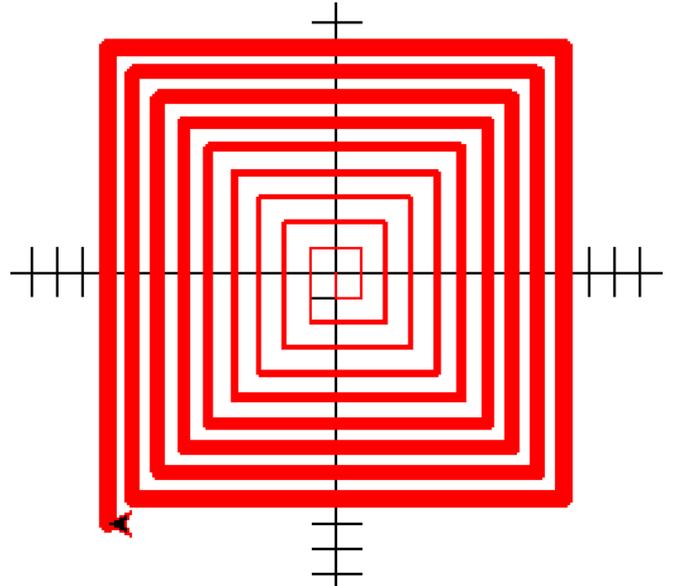
- Pour remplir un contour fermé, on utilise `begin_fill()` avant de tracer le contour fermé, puis `end_fill()` après.
- Pour définir la couleur, on utilise `fillcolor(0.2,0.2,0.4)` par exemple pour une couleur avec 20% de rouge, 20% de vert et 40% de bleu. Si les 3 coefficients sont égaux vous aurez des nuances de gris :
 - avec `fillcolor(0,0,0)` vous aurez du noir
 - avec `fillcolor(0.1,0.1,0.1)` vous aurez du gris foncé
 - avec `fillcolor(0.8,0.8,0.8)` vous aurez du gris clair
 - avec `fillcolor(1,1,1)` vous aurez du blanc

9. Tracer d'une spirale carrée :

⇒ Ecrire un code qui permette d'afficher la spirale ci-contre :

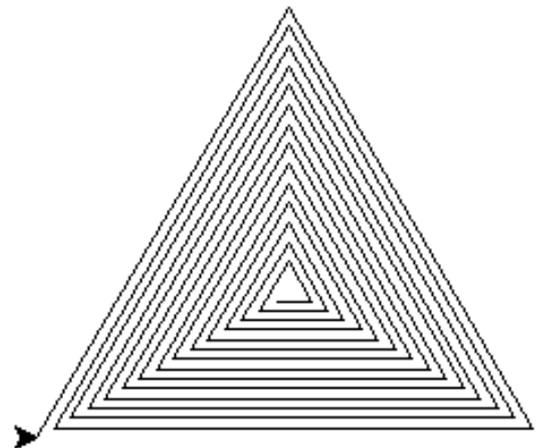
Le principe de cette spirale est de tracer des carrés à la suite, concentriquement et suivant la séquence suivante :

- Chaque carré est dessiné en traçant les côtes comme dans l'exercice 6
- Quand on passe d'un carré à l'autre la demi longueur est augmenté de 10 pixels
- Le coté vertical est plus long de dix pixels par rapport à la longueur du carré en cour, pour créer l'effet spirale
- On augmente d'un carré à l'autre l'épaisseur du trait de 0.8 pixel par la méthode *width()*



10. S'il reste du temps :

⇒ Ecrire un code qui permette d'afficher la figure suivante :



⇒ Ecrire un code qui permette d'afficher la figure suivante :

