

NOM et Prénom :

Vendredi 31 janvier 2025

1^{ère} – Spé N.S.I.

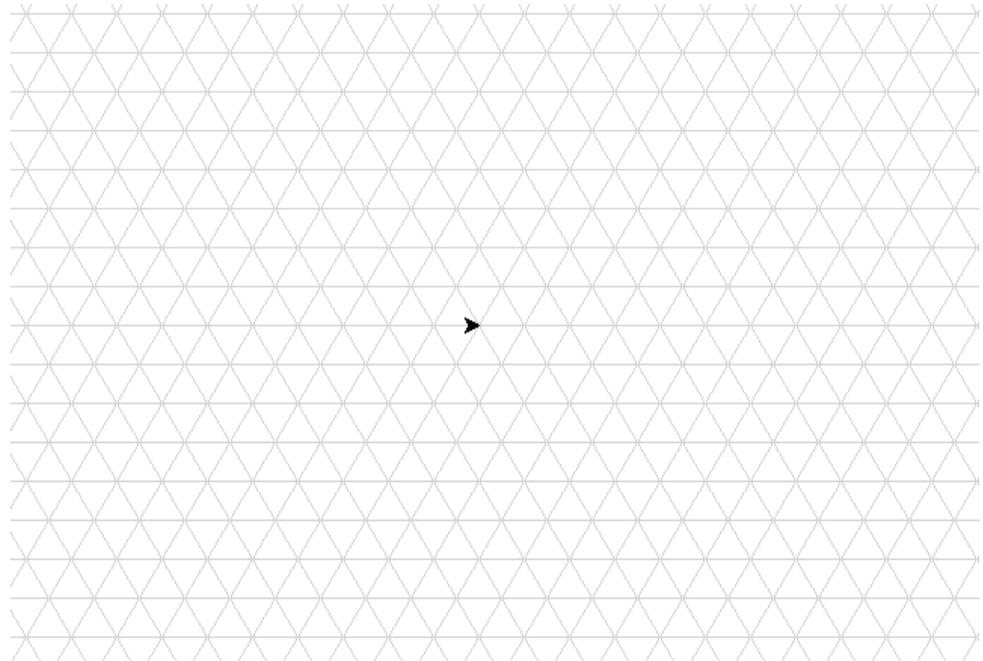
Interrogation (1h50)

(Calculatrice non autorisée)

Exercice 1 (4 points)

Dessiner la figure obtenue en exécutant le code suivant :

(Pour le quadrillage : le côté de chaque petit triangle équilatéral mesure 10 pixels)

<pre>import turtle as t def forme(n) : → for i in range(6): → → t.forward(n*10) → → t.left(60) → → for j in range(n) : → → → t.forward(10) → → → t.left(120) → → → t.forward(10) → → → t.right(120) → → t.right(180) → → t.forward(n*10) → → t.right(180) t.color("blue") forme(1) t.color("black") forme(4)</pre>	
--	---

Rappel :

b. Déplacer la tortue

- t.goto(x, y)* : Déplace la tortue vers le point de coordonnées (x, y) directement en ligne droite. Par défaut, l'origine du repère est au centre de la page.
- t.forward(n)* : Fait avancer la tortue de n pixels.
- t.backward(n)* : Fait reculer la tortue de n pixels.
- t.dot(r)* : Trace un point de rayon r pixels.
- t.circle(r, a)* : Trace un arc de cercle de rayon r pixels et d'angle a degrés. (Par défaut a vaut 360°)
Attention : La position de la tortue est le point le plus bas du cercle et non son centre !

c. Orienter la tortue

- t.left(a)* : Fait tourner la tortue sur elle-même de a degrés vers la gauche.
- t.right(a)* : Fait tourner la tortue sur elle-même de a degrés vers la droite.
- t.setheading(a)* : Définit l'orientation de la tortue vers a degrés sur le cercle trigo.

Exercice 3 (4 points)

Ecrire deux fonctions **verifier(i)** et **compter(n)** :

- **verifier(i)** : Le paramètre **i** étant un entier, la fonction renvoie un booléen.
 - **True** : si le chiffre 1 ou le chiffre 2 apparaît au moins une fois, mais qu'ils n'apparaissent pas tous les deux dans l'écriture de l'entier **i**.
 - **False** : sinon, c'est-à-dire qu'il n'y a ni le chiffre 1, ni le chiffre 2, ou qu'ils apparaissent tous les deux dans l'écriture de l'entier **i**.
- **compter(n)** : Le paramètre **n** étant un entier, la fonction renvoie un entier.
Nombre d'entiers compris entre 1 et **n** vérifiant la propriété précédente.

Exemples :

- `verifier(1)` renvoie **True**, car le chiffre 1 est présent.
- `verifier(2025)` renvoie **True**, car le chiffre 2 est présent.
- `verifier(1234)` renvoie **False**, car les chiffres 1 et 2 sont présents tous les deux.
- `verifier(6699)` renvoie **False**, car ni le chiffre 1, ni le chiffre 2 n'est présent.
- `compter(10)` renvoie **3**, car il y a les nombres 1, 2, et 10.
- `compter(15)` renvoie **7**, car il y a les nombres 1, 2, 10, 11, 13, 14 et 15.

Rappels :

- **str(objet)** : Permet de transformer un objet (un entier par exemple) en une chaîne de caractères.
- **ch_1 in ch_2** : Teste si la chaîne `ch_1` est incluse dans la chaîne `ch_2`.
Exemple : `"5" in "2025"` renvoie **True** alors que `"6" in "2025"` renvoie **False**.

def verifier(i) :	def compter(n) :
---	---

Exercice 4 (4 points)

Voici deux fonctions mystérieuses... **mystere_a(n)** et **mystere_b(n)** :

<pre>def mystere_a(n) : → if n%2 == 0 : → → return n//2 → else : → → return 3*n + 1</pre>	<pre>def mystere_b(n) : → i = 0 → u = n → while u != 1 : → → i += 1 → → u = mystere_a(u) → return i</pre>
---	---

Que renvoient les appels suivants :

mystere_a(5)
mystere_a(12)
mystere_b(8)
mystere_b(12)

Exercice 5 (4 points)

Ecrire la fonction **premier(n)** qui teste si le nombre **n** est premier en suivant l’algorithme indiqué, la fonction doit donc renvoyer un booléen :

- **True** : Si **n** est un nombre premier
- **False** : Si **n** n’est pas un nombre premier

<p>Définition de la fonction premier(n)</p> <p>Initialisation de la variable d à la valeur 2</p> <p>Tant que d² est inférieur ou égal à n</p> <p style="padding-left: 20px;">Si n est divisible par d,</p> <p style="padding-left: 40px;">Alors renvoyer Faux</p> <p style="padding-left: 20px;">Sinon</p> <p style="padding-left: 40px;">Alors ajouter 1 à la valeur d</p> <p>Renvoyer Vrai</p>	<p>.....</p>
--	---