

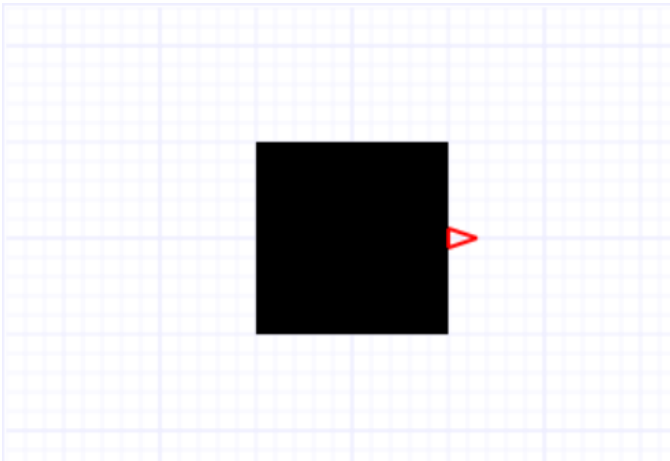
## Exercice noté #5

### Fonctions de dessins avec la Tortue

L'exercice consiste à concevoir des programmes qui définissent des procédures permettant de faire des dessins dans codeBoot avec la tortue. Donc dans votre codage vous devez vous servir (copier) tout le code utile créé précédemment et y ajouter ce qu'il faut pour réaliser la procédure demandée dans la question. Pour chaque programme, une procédure avec paramètres vous est demandée qui permet de dessiner une certaine forme et finalement, la dernière question vous demande d'animer une forme composée des formes précédemment créées. Toutes les formes de base sont toujours "solides" (il ne faut pas juste tracer le contour) et elles sont orientées à la position de la tortue. À la fin de l'exécution de n'importe laquelle de ces procédures la tortue doit être dans la position indiquée sur les images fournies. Pour chaque question vous devez terminer votre programme avec un appel précis à la procédure pour obtenir un dessin précis. Il n'y a pas de tests unitaires à faire pour ce travail car l'écriture de tests unitaires pour vérifier les dessins est problématique. Comme d'habitude votre programme doit être bien commenté et contenir des commentaires qui donnent votre nom, la date et des commentaires appropriés (utilité du programme, rôles des variables, explications de la logique du code, etc). Les identificateurs doivent être en camelCase. Le code ne doit pas contenir de caractères de tabulation. Bref, comme si votre code allait être utilisé par un autre programmeur dans le futur. Notez que l'espace pour éditer le code peut être agrandi en rapetissant la console (en glissant vers le haut la séparation entre la console et les fichiers), et aussi en réduisant la taille de la police utilisée par le navigateur web (normalement les touches Ctrl et "-" sur Linux et Command et "-" sur macOS). Vous pouvez aussi réduire la taille de la fenêtre de dessin en effectuant l'appel `clear(250, 150)` à la console (évitez de faire ça dans votre programme car ça va nuire à la correction).

### Question 1 (10 points)

Définir la procédure **carre(taille)** qui dessine un carré ayant la taille spécifiée d'un côté (un entier positif). Votre programme doit se terminer par l'appel **carre(100)** et doit afficher ce dessin

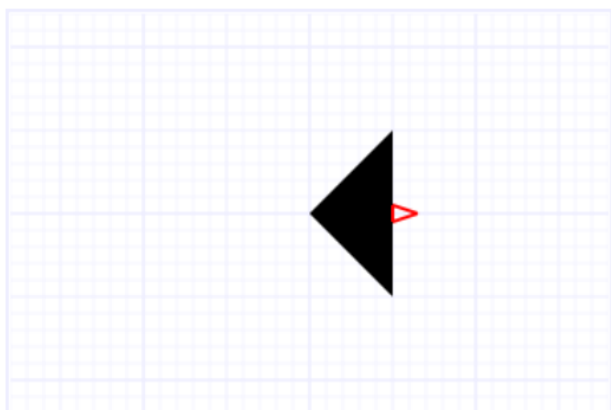


### Question 2 (20 points)

Définissez une procédure **triangle(base)** capable de dessiner un triangle isocèle avec une base de taille déterminée. Pour former un triangle comme une forme « solide », vous pouvez utiliser les segments placés étroitement pour le faire:

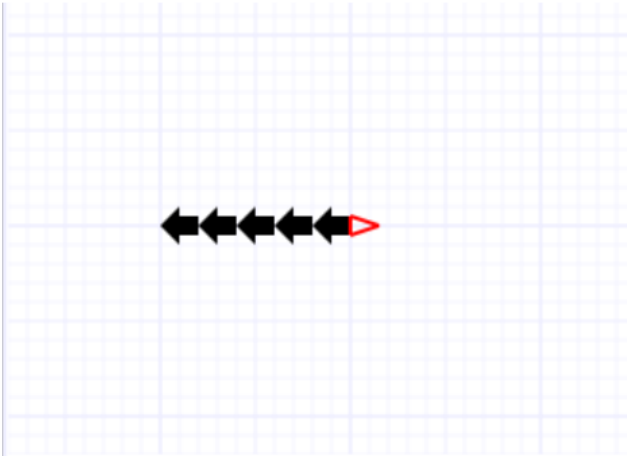


Le paramètre base est un nombre positif, et la taille des 2 autres cotés est un nombre positif arbitraire (mais doit être similaire au dessin). Votre programme doit se terminer par l'appel **triangle(100)** et doit afficher ce dessin :



### Question 3 (20 points)

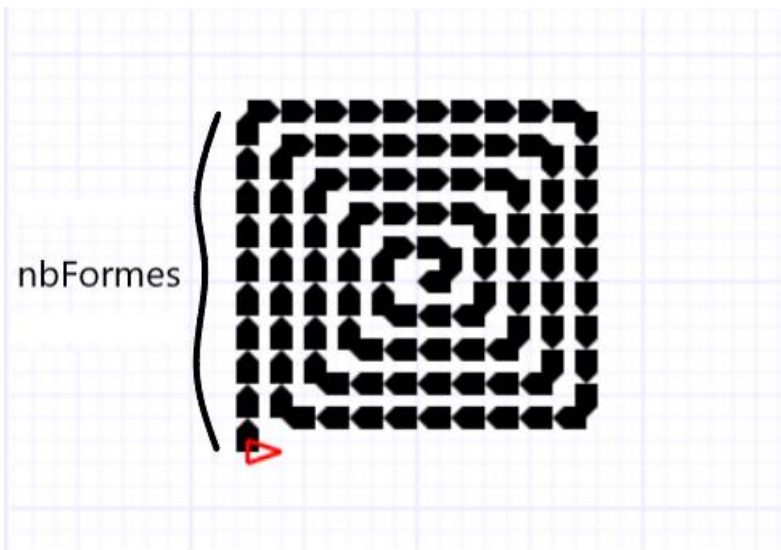
Votre programme doit définir la procédure **ligne(base, taille, n)** qui dessine une suite de  $n$  sous séquences altérant un triangle et un carré. Les carrés ont des cotés *taille* et les triangles *base* ont des bases *base*. Les deux - nombres positifs. La tortue doit être placée à la position  $(-100,0)$  avant de dessiner la ligne présentée plus bas. Votre programme doit se terminer par l'appel **ligne(20,10,5)** et doit afficher ce dessin :



### Question 4 (30 points)

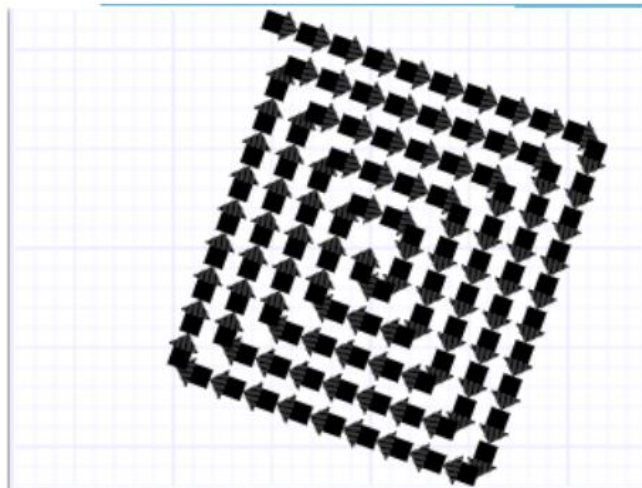
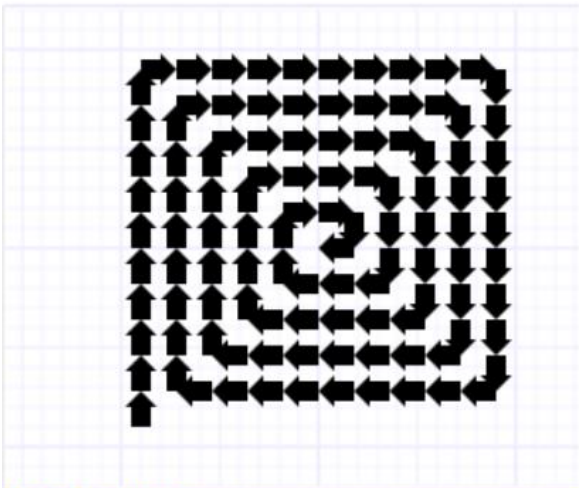
Votre programme doit définir la procédure **spirale(base, taille, nbFormes)** de 10 demi-tours qui dessine une spirale formée de « flèches ». Chaque « flèche » est composée d'une suite Triangle-Carré. Les tailles des formes composant une flèche sont spécifiées comme les deux premiers paramètres : *base*, *taille*.

**base** est un nombre positif qui spécifie la base du triangle et **taille** est un entier positif qui représente la taille du côté du carré, **nbFormes** spécifie le nombre de formes de la plus longue ligne de votre spirale. Votre programme doit se terminer par l'appel **spirale(10, 10, 10)** et doit afficher ce dessin:



### Question 5 (20 points)

Votre programme doit définir la procédure qui anime une spirale. **spiraleTournante(base, taille, nbFormes, vAng)** La spirale de 10 demi-tours est formée des lignes. Chaque ligne est composée de la séquence Triangle-Carré et la plus longue ligne est composée de **nbFormes** séquences. **taille** représente la taille de la forme Carré et **base** – la taille utilisée pour dessiner la forme Triangle. Les deux ces paramètres sont les entiers positifs. La rotation doit être effectuée avec la vitesse angulaire spécifiée comme le quatrième paramètre **vAng**, un entier. Si **vAng** est positive, la rotation doit être effectuée dans le sens horaire, sinon – dans le sens antihoraire. Votre programme doit se terminer par l'appel **spiraleTournante(15,10,10,2)** et doit tourner la spirale dans le sens horaire avec la vitesse angulaire 2 (tourner le dessin chaque 0.01 seconde à 2 degrés dans le sens horaire).



### Soumission :

Vous devez soumettre un seul fichier python avec toutes les fonctions déclarées puis appelées en commentaire en bas du script comme dans l'exemple ci-dessous :

```
#carre(100)
#triangle(100)
#ligne(20,10,5)
#spirale(10, 10, 10)
#spiraleTournante(15,10,10,2)
```